JOURNAL OF ALGORITHMS **29,** 1–17 (1998)
ARTICLE NO. AL980946

The k-Steiner Ratio in the Rectilinear Plane

Al Borchers* and Ding-Zhu Du†

Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455

Biao Gao‡

Lattice Semiconductor Corporation, 1820 McCarthy Blvd., Milpitas, California 95125

and

Pengjun Wan§

Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois 60616

Received April 14, 1995; revised October 16, 1997

A Steiner minimum tree (SMT) in the rectilinear plane is the shortest length tree interconnecting a set of points, called the regular points, possibly using additional vertices. A k-size Steiner minimum tree (kSMT) is one that can be split into components where all regular points are leaves and all components have at most k leaves. The k-Steiner ratio in the rectilinear plane, ρ_k, is the infimum of the ratios SMT/kSMT over all finite sets of regular points. The k-Steiner ratio is used to determine the performance ratio of several recent polynomial-time approximations for Steiner minimum trees. Previously it was known that in the rectilinear plane, $\rho_2 = 2/3$, $\rho_3 = 4/5$, and $(2k-2)/(2k-1) \le \rho_k(L_1) \le (2k-1)/(2k)$ for $k \ge 4$. In 1991, P. Berman and V. Ramaiyer conjectured that in fact $\rho_k = (2k-1)/(2k)$ for $k \ge 4$. In this paper we prove their conjecture.

* E-mail: borchers@cs.umn.edu. Supported in part by NSF grant CCR-95-30306.
† E-mail: dzd@cs.umn.edu. Also from Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China. Supported in part by NSF grant CCR-95-30306.
‡ E-mail: biao@lscsjc.lathicesemi.com.
§ E-mail: wan@csam.iit.edu.

0196-6774/98 $25.00

1. INTRODUCTION

Given a set of points in the rectilinear plane, a *Steiner minimum tree* is the shortest network interconnecting all the points. Those points in the set are called *regular points* and vertices other than regular points in the network are called *Steiner points*. Clearly the network will be a tree on the regular points and Steiner points.

The Steiner tree problem can also be considered in weighted graphs, in the Euclidean plane, and in any metric space. Computing a Steiner minimum tree in the rectilinear plane is NP-hard [9], as it is in graphs and the Euclidean plane [8, 7].

A tree interconnecting a regular point set is called a *Steiner tree* if every leaf is a regular point. However, a regular point in a Steiner tree may not be a leaf. A Steiner tree is *full* if every regular point is a leaf. When a regular point is not a leaf, the tree can be decomposed into several small trees. In this way, every Steiner tree can be decomposed into smaller trees where in each tree every regular point is a leaf. These smaller trees are called *full components* of the tree. The *size* of a full component is the number of regular points in the full component.

A k-size Steiner tree is a Steiner tree with all full components of size at most k. A k-size Steiner minimum tree is the shortest one among all k-size Steiner trees. The 2-size Steiner minimum tree is also called the minimum spanning tree. The k-Steiner ratio in a metric space E is defined by

$$\rho_k = \inf_{P \subset E} \frac{L_S(P)}{L_{kS}(P)},$$

where $L_S(P)$ is the length of a Steiner minimum tree for P and $L_{kS}(P)$ is the length of the k-size Steiner minimum tree for P. The 2-Steiner ratio is simply called the Steiner ratio. In the rectilinear plane, $\rho_2 = 2/3$ [12]. The Steiner ratios in graphs [10, 13] and in the Euclidean plane [5] have also been determined.

The k-Steiner ratio is important because of Steiner tree approximation algorithms. It was a long-standing open problem [3] whether there exists a polynomial-time approximation for a Steiner minimum tree in each metric space with performance ratio smaller than the inverse of the Steiner ratio. (The performance ratio of an approximation algorithm is the smallest upper bound for the ratio of lengths between the approximate solution and a Steiner minimum tree for the same set of points.) Zelikovsky [14] made the first breakthrough. He found a polynomial-time approximation for a Steiner minimum tree with performance ratio $(\rho_2^{-1} + \rho_3^{-1})/2$. By extending Zelikovsky's idea, Berman and Ramaiyer [1] gave a polynomial-time approximation for a Steiner minimum tree with performance ratio

$$\rho_2^{-1} - \frac{\rho_2^{-1} - \rho_3^{-1}}{2} - \frac{\rho_3^{-1} - \rho_4^{-1}}{3} - \cdots - \frac{\rho_{k-1}^{-1} - \rho_k^{-1}}{k-1},$$

and Du, Zhang, and Feng [6] showed a generalization of Zelikovsky's approximation with performance ratio

$$\frac{(k-2)\rho_2^{-1} + \rho_k^{-1}}{k-1}.$$

Recently, Zelikovsky [15] gave a new polynomial-time approximation for a Steiner minimum tree with performance ratio

$$\rho_k^{-1}(1 - \ln \rho_2 + \ln \rho_k).$$

Clearly, establishing a lower bound for the k-Steiner ratio is an important part for determining the performance ratio of these approximation algorithms. A better lower bound will give a better performance ratio for their approximations.

In graphs, the exact value of the k-Steiner ratio has been determined [4]. In the rectilinear plane, Berman and Ramaiyer [1] proved that $\rho_3 = 4/5$ and for $k \geq 4$, $\rho_k \geq (2k-2)/(2k-1)$. (See [2] for an excellent survey.) They also conjectured that for $k \geq 4$, $\rho_k = (2k-1)/(2k)$. In this paper we prove this conjecture.

THEOREM 1. *In the rectilinear plane,*

$$\rho_k = \frac{2k-1}{2k} \quad \textit{for } k \geq 4.$$

Our proof is based on a classical result of Hwang [12] about full rectilinear SMTs. Basically, we show how to break a full rectilinear SMT into a kST without increasing the length by more than a factor of the inverse of the k-Steiner ratio. There are two cases, k odd and k even. Although the general ideas are similar, the even and odd cases differ in details, and the even case is more complicated and involves more interesting techniques.

2. PRELIMINARIES

We begin with a result by Hwang [12] which says we can assume in the SMT that the full components on more than four regular points are in one of the two forms shown in Fig. 1. We label the regular points B_i for

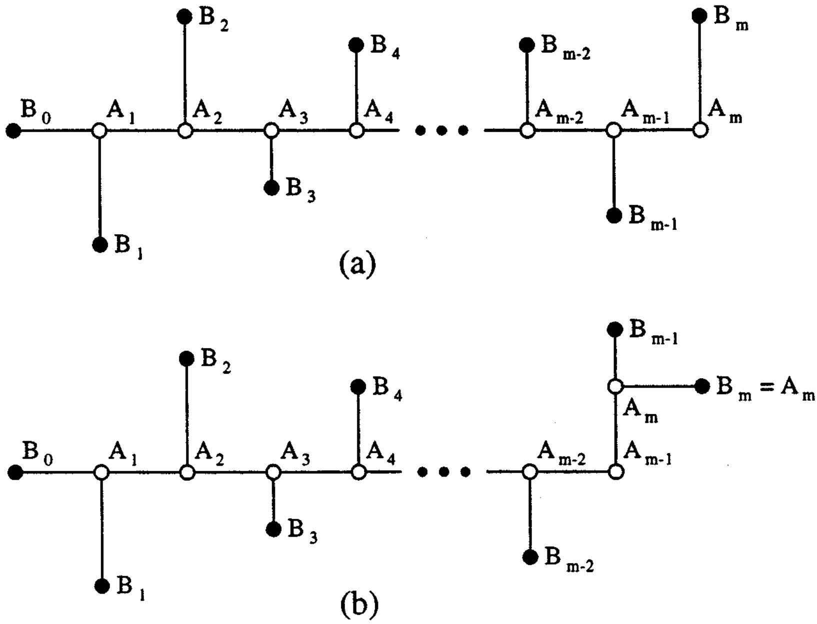

FIG. 1. Forms of a full component.

$0 \le i \le m$ and the Steiner points A_i for $1 \le i \le m$. Let $A_0 = B_0$, $x_i = |\overline{A_iB_i}|$ for $0 \le i \le m$, and $y_i = |\overline{A_{i-1}A_i}|$ for $1 \le i \le m$, where $|\overline{AB}|$ is considered as the rectilinear distance between A and B. Although x_i is the length of the vertical segment $\overline{A_iB_i}$, we will also use x_i to refer to the segment itself. Similarly, y_i refers to the length of a horizontal segment and the segment itself. The line along the Steiner points $A_0, A_1, \ldots$ is called the *spine* of the component.

We will use several basic transformations. Doubling at B_j, or doubling x_j, means duplicating the segment x_j and splitting the component at A_j to create two full components joined at B_j. Their total length is the length of the original component plus x_j. See Fig. 2.

A horizontal doubling of y_j is done as shown in Fig. 6 or Fig. 12 below. There are several cases, depending on the lengths of the vertical segments. In all cases the component is split into two or three components whose total length is increased only by y_j. For a full component in the form of Fig. 1(b), y_{m-1} is doubled as shown in Fig. 3.

Another simple transformation is shown in Fig. 3. If there is an x_j on the same side of the spine as x_m with $x_j < x_l$ for $j + 1 \le l \le m$, then we can slide the spin down from A_j to A_m and split the component into two

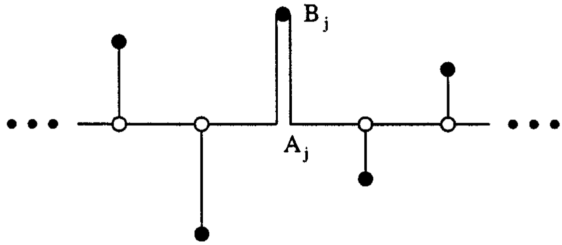

FIG. 2. Doubling x_j.

full components joined at B_j. Note that this does not change the length, unless B_j is the first regular point in a nonstandard component, in which case the length decreases by x_j. We can similarly slide the spine at the left end of a component.

LEMMA 1. *Upper bound*:

$$\rho_k \leq \frac{2k-1}{2k}.$$

Consider $k+1$ points in a full Steiner tree, as in Fig. 5, where all segments are of length 1. For these points the SMT is of length $2k-1$. It seems apparent that the length of a kSMT on these $k+1$ points must be at least $2k$. If so, this shows $(2k-1)/(2k)$ is an upper bound on ρ_k.

To justify this, we first use a theorem of Hanan [11] that states in the rectilinear plane, for any set of regular points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, there is a SMT where the Steiner points are all of the form (x_i, y_i) for some i and j. So all segments will be of integral length. The kSMT of these points must split into at least two components, say they are of size $k_1, k_2, \ldots, k_p$. Now each regular point in each component has degree exactly 1, so we count one segment of length 1 for each of the $k_1 + k_2 \cdots + k_p = k + p \geq k + 2$ points. Then in the whole kSMT, points B_i for $1 \leq i \leq k-2$ must

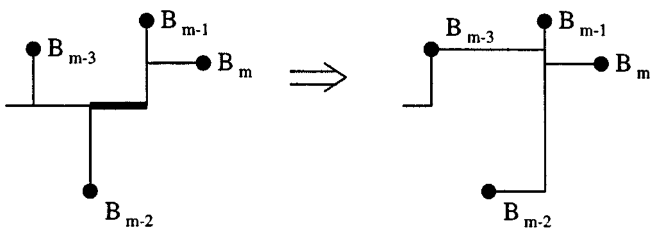

FIG. 3. Doubling y_{m-1} in the alternate form of a full component.

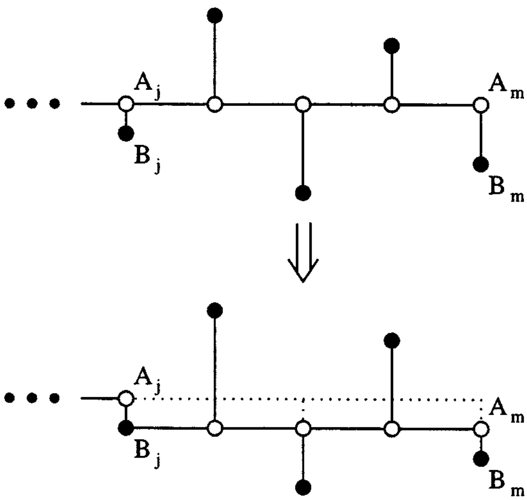

FIG. 4. Sliding the spine.

be connected to some point to their right, and these points are all at least distance 3 away; so we must count at least one more segment of length 1 for these points, for an additional length of $k - 2$. So the total length of the kSMT is at least $2k$.

Now we begin the lower bound proof. It is a proof by induction. When the number of regular points, $|P|$, is at most k, the SMT and the kSMT are the same, so $L_S(P)/L_{kS}(P) = 1 > (2k - 1)/(2k)$ and the lower bound holds. If $|P| > k$ and the SMT on P is not a full SMT, then we can split the SMT into two trees at a regular point of degree more than 1. By the induction hypothesis, the lower bound holds on each of the smaller sets of regular points. The length of the SMT is the sum of the lengths of the two smaller SMTs, and the length of the kSMT is at most the sum of the lengths of the kSMTs on the two smaller point sets. By a simple calculation it can be seen that the lower bound holds on all of P.

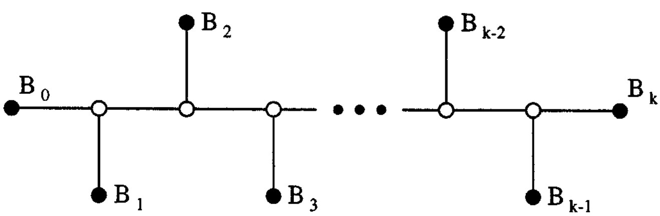

FIG. 5. The upper bound for ρ_k.

So we can assume that T is a full Steiner tree on regular points P, and we need only prove the lower bound in this situation. We will show that from T we can construct $2k-1$ k-restricted Steiner trees (kSTs) whose total length is at most $2k$ times the length of T. From this the lower bound will follow. We consider k odd first, then k even.

3. THE LOWER BOUND FOR k ODD

THEOREM 2. *If T is a full SMT on P and $k > 4$ is odd, then*

$$\frac{L_S(P)}{L_{kS}(P)} \geq \frac{2k-1}{2k}.$$

From T we must construct $2k-1$ kSTs whose length is at most $2k$ times the length of T. In the first phase we will construct k trees, T_i^1, by doubling the horizontal segments y_j, for $j \equiv i \pmod k$ and $i = 1, 2, \ldots, k$; this construction will be similar to that given by Berman and Ramaiyer except in some situations we will also need to double a vertical segment. In the second phase we will construct $k-1$ additional trees, T_i^2 for $i = 1, 2, \ldots, k-1$, by doubling vertical segments selected $(\operatorname{mod} k-1)$, but skipping over those segments already doubled in the first phase.

To construct T_i^1 we perform the transformation shown in Fig. 6, which doubles the segment y_j when $j \equiv i \pmod k$. Unless we have Case 1 occur at $j-k$ and at j (or at the right-hand end of the first component when $i = j = k$), then the components constructed in this way will have at most k regular points. This is easy to check. Consider the situation where we have a component, $\mathscr{C}_1$, with Case 1 at each end and so with $k+1$ regular points. See Fig. 7.

If there is a B_l in $\mathscr{C}_1$ on the same side of the spine as B_j with $x_l \leq x_j$, then we can slide the spine up between the rightmost such B_l and B_j and split the component at B_l into two components each of size less than k. So we can assume that $x_j < x_{j-2}, x_{j-4}, \ldots, x_{j-k+1}$.

Similarly, if there is a B_l in $\mathscr{C}_1$ on the same side as B_{j-k-1} (which is on the same side as B_j) with $x_l \leq x_{j-k-1}$, then we can slide the spin up between the leftmost such B_l and B_{j-k-1} and split the component at B_l into two components each of size less than k. So we can assume that $x_{j-k-1} < x_{j-2}, x_{j-4}, \ldots, x_{j-k+1}$.

Now consider the transformation shown in Fig. 8. Note that by our assumption, $x_j < x_{j-2}$, so this transformation is possible. The horizontal segment y_j is still doubled. Now the component $\mathscr{C}_1$ has k regular points, but the next component to the right, $\mathscr{C}_2$, has gained one new regular point.

If the $\mathscr{C}_2$ ends on the right in Case 2 of Fig. 6, then it has only k regular points; so assume it ends in Cases 1 or 3. Then B_{j+k-2} and B_{j+k} are in $\mathscr{C}_2$.

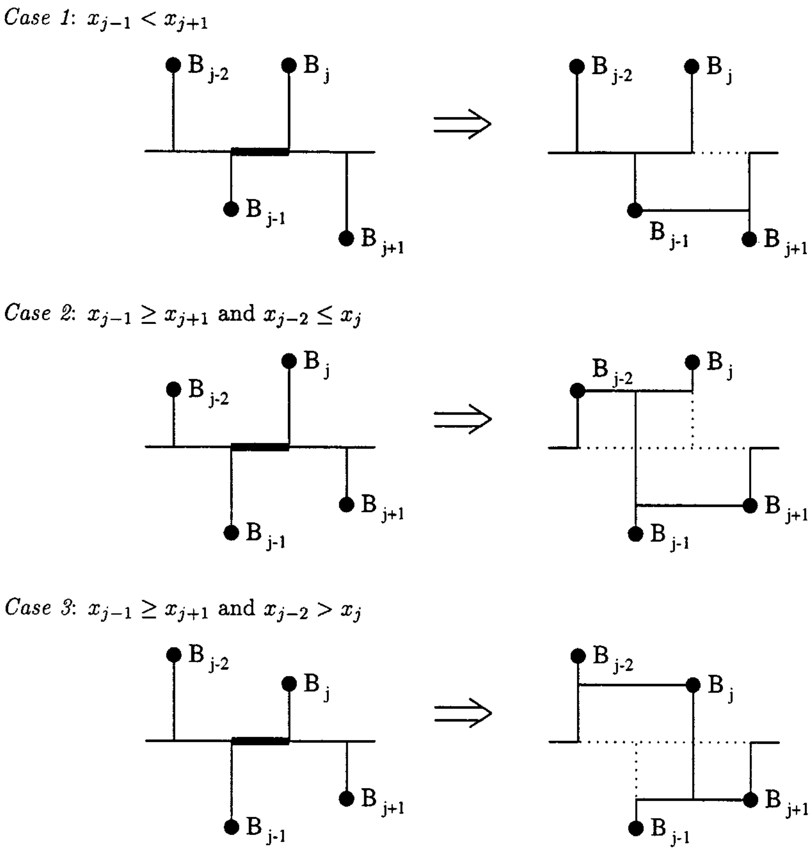

FIG. 6. Doubling y_j.

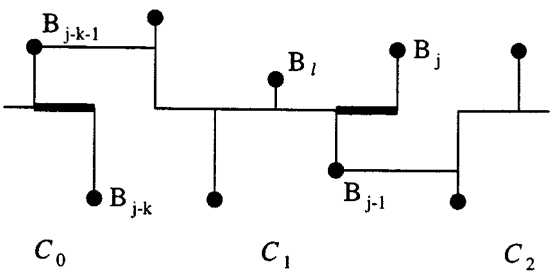

FIG. 7. A component with $k + 1$ regular points.

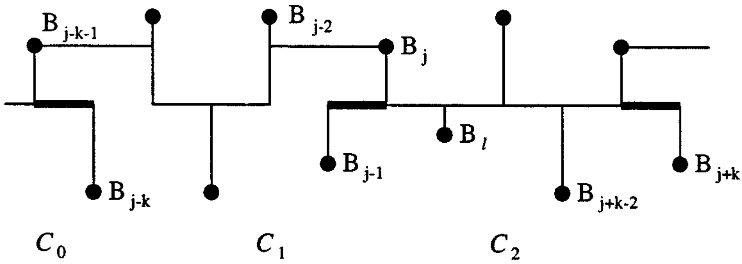

FIG. 8. Alternate doubling of y_j.

If there is a B_l in $\mathscr{C}_2$ on the same side of the spine as B_{j-1} with $l = j+1, j+3, \dots, j+k-2$ and with $x_l \le x_j$, then we can slide the spine down between B_{j-1} and the leftmost such B_l and split component $\mathscr{C}_2$ into two components, each of size at most k. So we can assume that $x_{j-1} < x_{j+1}, x_{j+3}, \dots, x_{j+k-2}$.

Similarly, we can assume that $x_{j+k} < x_{j+1}, x_{j+3}, \dots, x_{j+k-2}$.

Finally, consider the transformation shown in Fig. 9. Here we had to double the vertical segment x_{j-1}. In this situation, we mark both x_{j-1} and x_j so that they will not be doubled in the next phase, where we double vertical segments. This splits $\mathscr{C}_1$ into two components, one of size k and one of size 2, and it leaves $\mathscr{C}_2$ unchanged. We show the marked segments in bold.

When x_{j-1} and x_j are marked, we know

$$x_j, x_{j-k-1} < x_{j-2}, x_{j-4}, \dots, x_{j-k+1}$$

and

$$x_{j-1}, x_{j+k} < x_{j+1}, x_{j+3}, \dots, x_{j+k-2}.$$

As a consequence $x_{j-2}, x_{j-4}, \dots, x_{j-k+1}$ cannot also be marked, since a marked segment must be less than the $(k-1)/2$ vertical segments on the

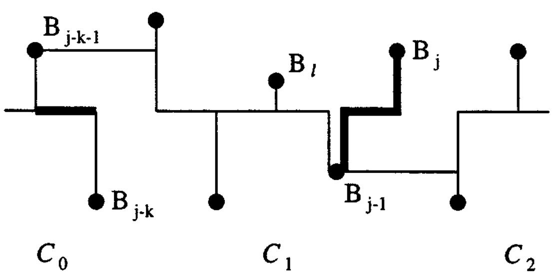

FIG. 9. Doubling and marking vertical segments.

same side of the spine, either to its left or right, but these segments are greater than both x_j and x_{j-k-1}. Similarly, $x_{j+1}, x_{j+3}, \ldots, x_{j+k-2}$ cannot be marked. In fact, since marked segments come in pairs, none of the segments from x_{j-k+1} to x_{j+k-2} can be marked besides x_{j-1} and x_j. So between any two marked segments there are at least $k-2$ unmarked vertical segments.

Now we begin the second phase. First we renumber the vertical segments, skipping any marked segments from the first phase. Using this new numbering, we construct T_i^2 by doubling at vertical segments x_j when $j \equiv i \pmod{k-1}$ for $i = 1, 2, \ldots, k-1$.

If a component constructed in this way has no marked segments, then it has at most k regular points.

If a component constructed in this way has a single pair of marked segments, then the component has $k+2$ regular points. Since k is odd, the end vertical segments of the component are both on the same side. Consider the marked segment on the same side as the end segments. The marked segment must be less than the $(k-1)/2$ vertical segments on its side of the spine, either to the left or right; but this must include one of the end segments. This follows because there are $(k+1)/2$ segments on this side of the spine and at least one on each side of the marked segment; so at most $(k-1)/2$ are to its left or right on the same side. Therefore, we can split the component at the marked segment into two components, each with at most k regular points, by sliding part of the spine. This is shown in Fig. 10(a).

Finally, it is possible that a component in T_i^2 has two pairs of marked segments and so $k+4$ regular points. If this happens, one pair must be the second and third vertical segments of the component and the other pair the second to last and third to last vertical segments. By the conditions on marked segments, we know that the first segment must be larger

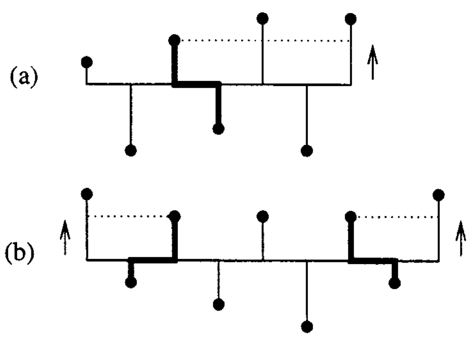

FIG. 10. Splitting components after phase two.

than the third and the last segment larger than the third to last, so we can split the component at each end into two components of size 3 and one component in the middle of size k. See Fig. 10(b).

Throughout this construction we have assumed that T is of the form shown in Fig. 1(a). If instead T is of the form shown in Fig. 1(b), then we first transform T by sliding down $\overline{A_m B_m}$ until A_m and A_{m-1} coincide. The transformed T is now of form Fig. 1(a) and we can obtain the k-restricted Steiner trees as we have done above. The length of these trees does not increase when we slide $\overline{A_m B_m}$ back to its correct place.

We have constructed from T, $2k-1$ kSTs, k in the first phase and $k-1$ in the second phase. Together these kSTs use each segment of T at most $2k$ times, once in each kST and once when the segment is doubled, which happens at most once in all the kSTs. Therefore, one of the kSTs, T', must have length at most $(2k)/(2k-1)$ times the length of T. Since $L_{kS}(P) \leq \text{length}(T')$ we get

$$\frac{L_S(P)}{L_{kS}(P)} \geq \frac{\text{length}(T)}{\text{length}(T')} \geq \frac{2k-1}{2k}.$$

4. THE LOWER BOUND FOR k EVEN

From T we will construct $2k-1$ kSts whose length is at most $2k$ times the length of T. In the first phase we will construct the trees T_i^1 for each $i = 1, 2, \ldots, k$, by doubling the vertical segments x_j in decreasing order of j for $j \equiv i \pmod k$; in some situations we will actually double horizontal segments and mark them either red or blue. In the second phase we will construct the trees T_i^2 for each $i = 1, 2, \ldots, k-1$, by doubling horizontal segments selected $(\bmod\, k-1)$, but skipping over those segments marked in the first phase; in some situations we will actually double vertical segments undoubled in the first phase. We will use the following algorithm to construct T_i^1. Let m_j be the minimum of $x_j, x_{j+2}, \ldots, x_{j+k}$.

ALGORITHM

(1) mark each x_j with $j \equiv i \pmod k$;

(2) **for each** marked x_j in decreasing order of j **do**

(3) unmark x_j;

(4) **if** $j + k < m$, or $\max\{x_j, x_{j+k}\} \geq \min\{x_{j+2}, x_{j+4}, \ldots, x_{j+k-2}\}$,

(5) double x_j as in Fig. 11;

(6) slide the spine in the component to split it and shorten it by m_j;

(7) **else**

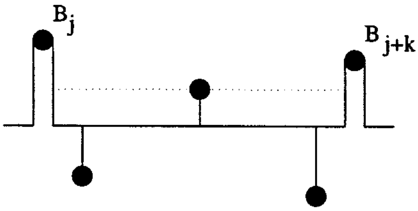

FIG. 11. Doubling x_j.

unmark x_{j-k} and double it;
if $x_{j+1} \geq x_{j-1}$,
  color y_{j+1} blue;
  double y_{j+1} as shown in Fig. 12;
  slide the spine in the right component to shorten it by m_j;
else
  color y_j red;
  if $x_j \geq x_{j-2}$,
    double y_j as shown in Fig. 6 Case 2;
  else
    double y_j as shown in Fig. 6 Case 3;
  slide the spine in the left component to shorten it by m_{j-k};

end-algorithm.

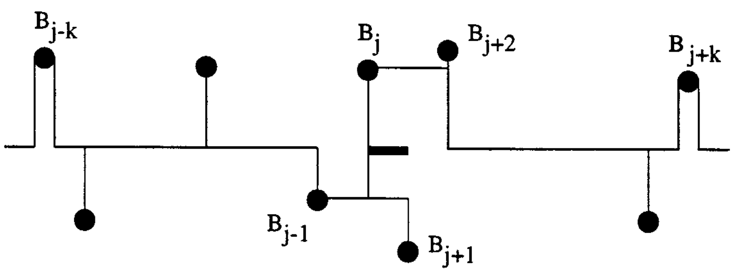

FIG. 12. Doubling y_{j+1} and coloring it blue.

We have the following several remarks on the algorithm, which are easy to verify:

Remark 1. All components constructed by the algorithm have at most k regular points.

Remark 2. If y_j is colored red, then $x_{j-1} > x_{j+1}$ which ensures that it cannot be colored in T^1_{i-1}. If y_{j+1} is colored with blue, then $x_j < x_{j+2}$ which ensures that it cannot be colored red in T^1_{i+1}. Therefore, each horizontal segment can be colored at most once.

Remark 3. Segment x_j is saved whenever y_j is red or y_{y+1} is blue. Segment m_j is saved except when y_{j+k+1} is blue or y_j is red.

Remark 4. If y_{j+1} is blue or y_j is red, then neither of y_{j-k+1} or y_{j+k+1} could be blue and neither of y_{j-k} or y_{j+k} could be red. Furthermore, since by step 4 of the algorithm, x_j and x_{j+k} must be shorter than the vertical segments on the same side of the spine between them, none of $y_{j+3}, y_{j+5}, \ldots, y_{j+k-1}$ could be blue and none of $y_{j+2}, y_{j+4}, \ldots, y_{j+k-2}$ could be red. Therefore, for two colored segments with the same color and the same parity or with different color and different parity, there are at least k horizontal segments between them.

Now we start the second phase. First we temporarily renumber the horizontal segments, skipping any colored segments from the first phase. Using this new numbering, we construct T^2_i by doubling the segments y_j when $j \equiv i \pmod{k-1}$ for $i = 1, 2, \ldots, k-1$, as shown in Fig. 6. For the rest of the proof, we will again use the original numbering of the regular points.

If the components constructed in this way have no colored segments, they have at most k regular points. Components could also contain one colored segment and at most $k+1$ regular points or two colored segments and at most $k+2$ regular points.

It is impossible for a component to have three colored segments. If it did, it would cover $k+3$ horizontal segments and the end segments would not be colored, so the colored segments could be separated by at most $k-1$ other horizontal segments. Of the three colored segments, either two would be the same color and same parity or two would be of different color and different parity; but this is not possible by Remark 4.

Suppose now that a component contains exactly one colored segment. The only case in which such a component would have more than k regular points is when Case 1 of Fig. 6 occurs at both ends when doubling y_j and y_{j+k}. In this case, we can assume that $x_{j+k} < x_{j+2}, x_{j+4}, \ldots, x_{j+k-2}$; otherwise we can slide the spine and split the component. If $x_{j+k-1} \geq \min\{x_{j+k+1}, x_{j+k+3}, \ldots, x_{j+2k-3}\}$, we can double y_{j+k} as in Fig. 13, which

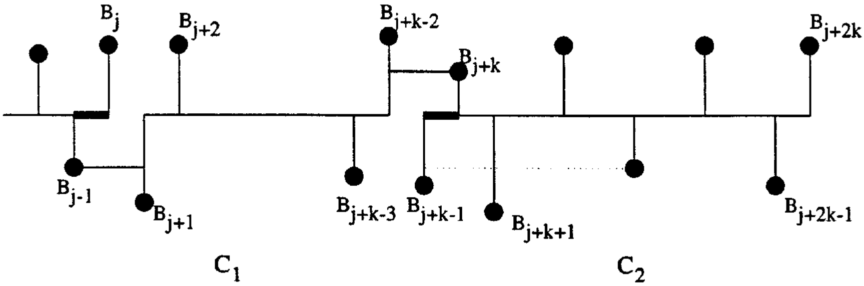

FIG. 13. Doubling y_{j+k} in another way and sliding the spine in C_2.

shifts one regular point to component C_2 on the right, and then slide the spine in C_2 to split C_2 into two trees. Therefore, we can further assume that $x_{j+k-1} < x_{j+k+1}, x_{j+k+3}, \ldots, x_{j+2k-3}$.

By step 4 of the algorithm, if y_j is colored red or y_{j+1} is colored blue, then x_j and x_{j+k} must be smaller than all vertical segments between them and on the same side of the spine. But given the two inequalities from the preceding paragraph, we can conclude that the only possible colored segments in C_1 are y_{j+1} colored blue or y_{j+k-1} colored red.

So y_{j+k-3} is not red; if in addition y_{j+2k-2} is not blue, then by Remark 3, we know m_{j+k-3} was saved in phase one, and we can split C_1 by doubling the shorter of x_{j+k-1} and x_{j+k-3}, which is equal to m_{j+k-3}.

On the other hand, if y_{j+2k-2} is blue, then by Remark 4, y_{j+k-1} is not red and y_{j+2k} is not blue, so m_{j+k-1} was saved in phase one. Also, when y_{j+2k-2} is blue, $x_{j+2k-3} < x_{j+2k-1}$ and so $m_{j+k-1} = x_{j+k-1}$. Hence, we can double x_{j+k-1} and split C_1 into components of size 2 and k.

So in all cases in which a phase two component has one colored horizontal segment, it can be split into components of at most k regular points.

Next we consider the case where a component contains exactly two colored segments. To have more than k regular points we must have one of four configuration from Fig. 6: Case 1, 2, or 3 on the left and Case 1 on the right, or Case 1 on the left and Case 3 on the right.

Here we can assume x_{j+k+1}, $\min\{x_{j-1}, x_{j+1}\} < x_{j+3}, x_{j+5}, \ldots, x_{j+k-1}$; otherwise, we could slide the spine to split the component somewhere in the middle into components of size less than k. From this inequality and the constraints imposed by step 4 of the algorithm, we know that y_{j+2} is the only possible blue segment of that parity and y_{j+1} is the only possible red segment of that parity, and of course only one of these two is possible.

By Remark 4, we know the two colored segments in this component must be of the same color and different parity or of different color and the same parity. In either case, we must have either y_{j+2} blue or y_{j+1} red. So we know x_{j+1} was saved in the first phase. If the left end of our component has Case 1 from Fig. 6, we can now double x_{j+1} to split off one regular point. If the right side was Case 3, the remaining component has only k regular points and we are done. So we can assume the remaining component begins at regular point B_{j+1}, ends on the right at B_{j+k+1} in Case 1, and has $k + 1$ regular points.

As we did for components with just one colored segment, we can assume $x_{j+k} < x_{j+k+2}, x_{j+k+4}, \ldots, x_{j+2k-2}$; otherwise, we could use an alternate horizontal doubling of y_{j+k+1} similar to that shown in Fig. 13 to shift one regular point to the component on the right and slide the spine of that component to split it.

With this inequality, we know that y_{j+k} is the only possible red segment of that parity in this component and that there is no blue segment of the other parity. Hence, in fact, y_{j+k} must be colored red. Also, from this inequality, m_{j+k-2} is the smaller of x_{j+k-2} and x_{j+k}. Since y_{j+k} is red, y_{j+k-2} is not red and y_{j+2k-1} is not blue, so m_{j+k-2} has been saved in phase one. We can then double the smaller of x_{j+k-2} and x_{j+k}, which will split the component into two components of size at most k and 4. An example is shown in Fig. 14.

Hence a phase two component with two colored segments can always be split into components with at most k regular points.

Finally, we note that each x_j and m_j saved in phase one is used at most once in phase two. The vertical segment x_j will be doubled in phase two only when it is in a component begun by doubling y_{j-1}. The segment m_j will be doubled in phase two only in one of these three cases:

1. When the component begun by doubling y_{j-k+1} contains exactly one colored segment, either y_{j-k+2} blue or y_j red, and y_{j+k-1} is blue.

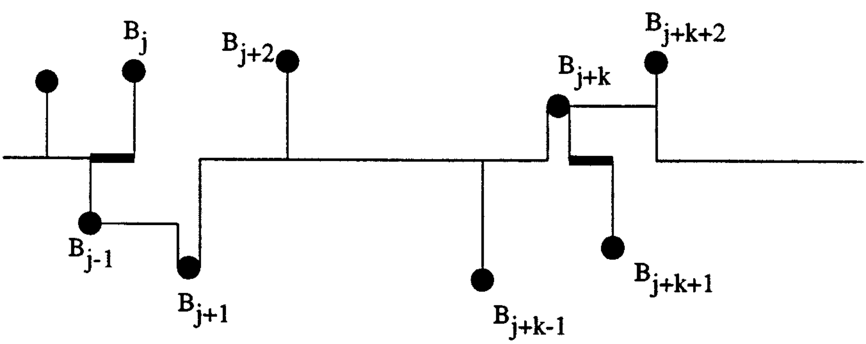

FIG. 14. Doubling x_{j+1} and x_{j+k} when a component has two colored segments.

2. When the component begun by doubling y_{j-k+3} contains exactly one colored segment, either y_{j-k+4} blue or y_{j+2} red, and y_{j+k+1} is not blue.

3. When the component begun by doubling y_{j-k+2} contains exactly two colored segments, y_{j+2} red and either y_{j-k+3} red or y_{j-k+4} blue.

We can see that the first two cases are mutually exclusive using Remark 4; we would have to have y_{j-k+2} blue, so y_{j+2} would be red, but then y_{j+k-1} could not be blue. The first and third cases are mutually exclusive again by using Remark 4, and the second and third are clearly mutually exclusive.

We conclude the proof for k even exactly as we did in the last paragraph of Section 3 for k odd.

5. DISCUSSION

Although the k-Steiner ratio in rectilinear plane and in graphs has been determined for every $k \geq 2$, the k-Steiner ratio in the Euclidean plane for $k \geq 3$ is still open. Du, Zhang, and Feng [6] conjectured that the 3-Steiner ratio in the Euclidean plane is $\rho_3(E^2) = (1 + \sqrt{3})\sqrt{2}/(1 + \sqrt{2} + \sqrt{3})$. If this is proved, we would obtain the best known performance ratio of $(2/\sqrt{3} + \rho_3^{-1}(E^2))/2$ for a polynomial-time approximation of the Euclidean SMT. However, it seems that studying the k-Steiner ratio in the Euclidean plane requires entirely different methods.

REFERENCES

1. P. Berman and V. Ramaiyer, Improved approximation algorithms for the Steiner tree problem, *J. Algorithms* **17** (1994), 381–408.
2. M. Bern and D. Eppstein, Approximation algorithms for geometric problems, *in* "Approximation Algorithms for NP-Complete Problems" (D. Hochbaum, Ed.), PWS–Kent, Boston, 1995.
3. M. Bern and P. Plassmann, The Steiner problem with edge lengths 1 and 2, *Infor. Process. Lett.* **32** (1989), 171–176.
4. Al Borchers and Ding-Zhu Du, The k-Steiner ratio in graphs, *SIAM J. Comput.* **26** (1997), 857–869.
5. D. Z. Du and F. K. Hwang, An approach for proving lower bounds: Solution of Gilbert–Pollak's conjecture on Steiner ratio, *in* "Proceedings of the 32nd Annual IEEE Symposium on the Foundations of Computer Science," 1990, pp. 76–85.
6. D. Z. Du, Y. J. Zhang, and Q. Feng, On better heuristic for Euclidean Steiner minimum trees, *in* "Proceedings of the 32nd Annual Symposium on the Foundations of Computer Science," 1991, pp. 431–439.

7. L. R. Foulds and R. L. Graham, The Steiner problem in phylogeny is NP-complete, *Adv. in Appl. Math.* **3** (1982), 43–49.
8. M. R. Garey, R. L. Graham, and D. S. Johnson, The complexity of computing Steiner minimal trees, *SIAM J. Appl. Math.* **32** (1977), 835–859.
9. M. R. Garey and D. S. Johnson, The rectilinear Steiner tree problem is NP-complete, *SIAM J. Appl. Math.* **32** (1977), 826–834.
10. E. N. Gilbert and H. O. Pollak, Steiner minimal trees, *SIAM J. Appl. Math.* **16** (1968), 1–29.
11. M. Hanan, On Steiner's problem with rectilinear distance, *SIAM J. Appl. Math.* **14** (1966), 255–265.
12. F. K. Hwang, On Steiner minimal trees with rectilinear distance, *SIAM J. Appl. Math.* **30** (1976), 104–114.
13. R. M. Karp, Reducibility among combinatorial problems, *in* "Complexity of Computer Computations," (R. E. Miller and J. W. Thatcher, Eds.), Plenum Press, New York, 1972, pp. 85–103.
14. A. Z. Zelikovsky, An 11/6-approximation algorithm for the network Steiner problem, *Algorithmica* **9** (1993), 463–470.
15. A. Z. Zelikovsky, Better approximation bounds for the network and Euclidean Steiner tree problems, manuscript.