Probabilistic Analysis of Rectilinear Steiner Trees

Chunhong Chen

Department of Electrical and Computer Engineering
University of Windsor, Ontario, Canada, N9B 3P4
E-mail: cchen@uwindsor.ca

Abstract

Steiner tree is a fundamental problem in the automatic interconnect optimization for VLSI design. We present a probabilistic analysis method for constructing rectilinear Steiner trees. The best solution under statistical sense is obtained for any given set of N points. Experiments show that our results are better than those by the previous technique or very close to the optima.

1. Introduction

One of the key problems in VLSI interconnect design is the topology construction of signal nets with the minimum cost. The *Steiner tree* problem is to find the tree structure which connects all pins of the signal net such that the wire length (i.e., cost) can be minimized. If all edges of the tree are restricted to the horizontal and vertical directions as are the case in VLSI design, the problem is called *rectilinear Steiner tree* (RST). In general, the RST can contain, in addition to the pins of the net, some more points which are called *Steiner points*. In particular, the RST without Steiner points is called *rectilinear minimum spanning tree* (RMST) which has been well studied [1]. While the RST can lead to better results than the RMST in terms of wire length, it has been shown that the RST problem is NP-complete [2]. Several effective heuristic approaches have been proposed towards the optimal or sub-optimal solutions. For example, *Hanan* [3] showed an optimal algorithm when the net contains no more than four pins. *Cohoon* [4] proposed an optimal algorithm when the pins of a net lie on the perimeter of a rectangle. *Hwang* [6] proved that the ratio of tree lengths between an RMST and an RST is no worse than 3/2. An o(N log N) algorithm for the RST was also proposed in [5], while the results were far from the optimal solution. A good survey on Steiner tree problems can be found in [7]. For a comprehensive survey of the interconnect design, the readers are referred to [10].

In this paper we provide probabilistic analysis for the rectilinear Steiner tree problem. By considering all possible topologic structures connecting every pair of pins, we can calculate the probability of the structures passing over individual edges. The optimal Steiner tree under statistical sense is the tree with maximum sum of the probabilities for all edges of which the tree is comprised. Experiment shows that the obtained tree topology is very close to the optimal RST. In the next section, some background together with a probabilistic model is described. Then, we show the Steiner tree construction algorithm in Section 3, followed by the experiments given in Section 4. Finally, Section 5 concludes the paper.

2. Probabilistic model for rectilinear steiner trees

In this section, we introduce some preliminaries and propose a probabilistic model for rectilinear Steiner trees.

2.1. Grid graph

Consider a set of N points, $\boldsymbol{P} = \{p_1, p_2, \ldots, p_N\}$ in a plane, where the location of p_i is denoted by (x_i, y_i). Assuming $x_i \neq x_j$ and $y_i \neq y_j$ for $i \neq j$ (discussions will be given later on if this is not the case), we can construct a *grid graph* which consists of the intersections (or, *segments*) of horizontal and vertical lines through all points. It was shown [3] that only those segments within the smallest rectangle enclosing all points need to be considered in obtaining the RST. An optimal RST is a subset of segments, $\boldsymbol{T}$, such that $\boldsymbol{T}$ is a tree for given points and the total wire length over all segments in $\boldsymbol{T}$ is minimum. Figure 1 illustrates the grid graph for a set of three points, $\boldsymbol{P} = \{p_1, p_2, p_3\}$. An optimal Steiner tree of Figure 1 is shown in Figure 2, where S_1 is a Steiner point.

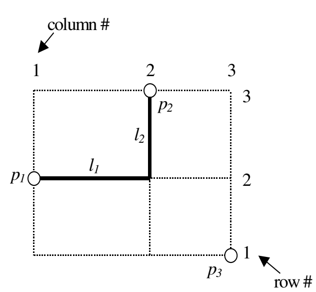


Figure 1. The grid graph for a set of three points

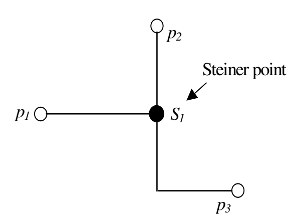


Figure 2. An optimal Steiner tree of Figure 1

If we number the columns and rows of the grid graph, the symbol $R(i,\ j)$ can be used to represent the horizontal segment which lies on row i and between columns j and j+1.

Similarly, we use $C(i, j)$ to represent the vertical segment which lies on column j and between rows i and $i + 1$. For instance, the segments l_1 and l_2 in Figure 1 can be denoted by $R(2, 1)$ and $C(2, 2)$, respectively. Note that the rows are numbered from bottom to top in the graph, and the columns are numbered from left to right. Before going further, we have the following definitions.

Definition 1: Given two points $p_i, p_j \in \boldsymbol{P}$, the distance $m = |c(i) - c(j)|$ is called the *horizontal grid-distance* between them , where $c(i)$ and $c(j)$ are the column numbers of p_i and p_j , respectively. Similarly, the *vertical grid-distance* between them is defined to be $n = |r(i) - r(j)|$, where $r(i)$ and $r(j)$ are the row numbers of p_i and p_j, respectively.

Definition 2: If $0 \le c(j) - c(i) \le 1$ for points $p_i, p_j \in \boldsymbol{P}$, then $H(k) = x_j - x_i$ is called the k-th *horizontal physical-length* of the grid graph, where $k = c(i)$. If $0 \le r(j) - r(i) \le 1$ for points $p_i, p_j \in \boldsymbol{P}$, then $V(l) = y_j - y_i$ is defined to be the l-th *vertical physical-length* of the graph, where $l = r(i)$.

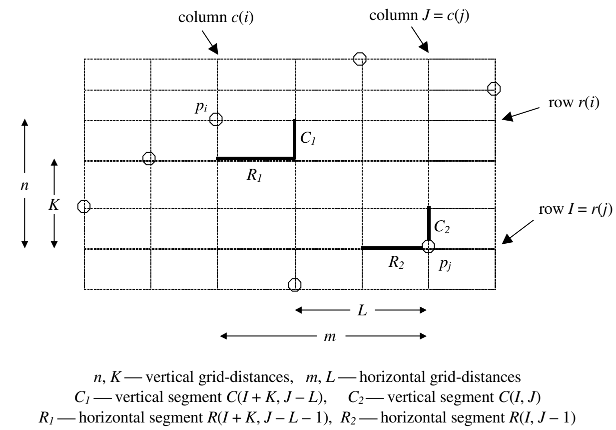


Figure 3. Probabilistic analysis for the segments through which the shortest paths between points (p_i and p_j) pass

2.2. Probabilistic model

Consider two points $p_i, p_j \in \boldsymbol{P}$ in the grid graph as shown in Figure 3. Without loss of generality, we assume $c(i) < c(j)$ and $r(i) > r(j)$. Let $I = r(j)$, and $J = c(j)$. The horizontal and vertical grid distances between p_i and p_j are $m = c(j) - c(i)$, and $n = r(i) - r(j)$, respectively. Let M be the number of all possible shortest paths from p_i to p_j . The number of those paths which pass through the segment $R(I, J - 1)$ (i.e., R_2 in Figure 3) only depends on m and n, and is denoted by $F(m, n)$. The number of those paths which pass through the segment $C(I, J)$ (i.e., C_2 in Figure 3) is also a function of m and n, denoted by $G(m, n)$. Obviously, we have $M = F(m, n) + G(m, n)$. In particular, for any positive integer q, we have $F(1, q) = G(q, 1) = 1$, and $F(q, 1) = G(1, q) = q$. From a statistical point of view, the probability of a shortest path between the two points passing through $R(I, J - 1)$ (or, $C(I, J)$) is given by $F(m, n)/M$ (or, $G(m, n)/M$). Furthermore, from Figure 3, $F(m, n)$ and $G(m, n)$ can be written as:

$$F(m,n) = F(m-1,n) + G(m-1,n) \quad (1)$$

and

$$G(m,n) = F(m,n-1) + G(m,n-1) \quad (2)$$

or,

$$F(m,n-1) = F(m-1,n-1) + G(m-1,n-1) \quad (3)$$

and

$$F(m-1,n-1) = G(m-1,n) - G(m-1,n-1) \quad (4)$$

Theorem 1: If we define $F(q, 0) = G(0, q) = 1$ for any integer $q > 0$, then $F(m, n)$ and $G(m, n)$ can be computed recursively as follows:

$$F(m,n) = \sum_{k=0}^{n} F(m-1,k)$$

$$G(m,n) = \sum_{k=1}^{n} G(m-1,k)$$

and

$$M = F(m+1,n) = G(m,n+1)$$

where $m, n > 1$, and $F(m, n)$, $G(m, n)$, and M are defined as earlier.

Proof : The second part of the theorem is straightforward, and we prove the first part only. Adding equations (3) and (4) gives

$$F(m,n-1) = G(m-1,n) \quad (5)$$

or

$$F(m+1,n-1) = G(m,n) \quad (6)$$

From (1) and (5), we have

$$\begin{aligned} F(m,n) &= F(m-1,n) + F(m,n-1) \\ &= F(m-1,n) + F(m-1,n-1) + F(m,n-2) \\ &= \cdots = F(m,1) + \sum_{k=2}^{n} F(m-1,k) \end{aligned} \quad (7)$$

Since $F(m, 1) = m$ and $F(q, 0) = 1$, equation (7) can be rewritten as

$$F(m,n) = \sum_{k=0}^{n} F(m-1,k) \quad (8)$$

From (6) and (8), we have

$$G(m,n) = F(m+1, n-1) = \sum_{k=0}^{n-1} F(m,k) = \sum_{k=0}^{n-1} G(m-1,k+1) = \sum_{k=1}^{n} G(m-1,k) \quad (9)$$

■

The values of $F(m, n)$ and $G(m, n)$ for $m, n \le 6$ are shown in Table 1 and Table 2, respectively.

Table 1. $F(m, n)$

m \ n	1	2	3	4	5	6
1	1	1	1	1	1	1
2	2	3	4	5	6	7
3	3	6	10	15	21	28
4	4	10	20	35	56	84
5	5	15	35	70	126	210
6	6	21	56	126	252	462

Table 2. $G(m, n)$

m \ n	1	2	3	4	5	6
1	1	2	3	4	5	6
2	1	3	6	10	15	21
3	1	4	10	20	35	56
4	1	5	15	35	70	126
5	1	6	21	56	126	252
6	1	7	28	84	210	462

More generally, let us consider the specific horizontal segment $R(I + k, J - l - 1)$ in Figure 3, where $0 \le k \le n$, $0 \le l \le m - 1$. Among all M shortest paths from p_i to p_j, the number of paths passing through this segment is given by

$$F(m - l, n - k)\cdot[F(l, k) + G(l, k)] = F(m - l, n - k)\cdot F(l + 1, k)$$

Thus, the probability of a shortest path passing through this segment is expressed as

$$PR(I+k, J-l-1) = \frac{F(m-l, n-k)\cdot F(l+1,k)}{F(m+1,n)} \quad (10)$$

Similarly, the probability of a shortest path between p_i and p_j passing through the specific vertical segment $C(I + k, J - l)$ (refer to Figure 3) is expressed as

$$PC(I+k, J-l) = \frac{G(m-l, n-k)\cdot G(l,k+1)}{G(m,n+1)} \quad (11)$$

where $0 \le k \le n - 1$, and $0 \le l \le m$.

When we account for the shortest paths for all $N\cdot(N - 1)/2$ pairs of points in the grid graph, two *probability matrices*, (denoted by ***PR*** and ***PC***), can be defined to represent the probabilities of all horizontal and vertical segments, respectively, through which the shortest paths would pass. The element $PR(i, j)$ in ***PR*** (or $PC(i, j)$ in ***PC***) corresponds to a horizontal (or vertical) segment $R(i, j)$ (or $C(i, j)$) in the graph. ***PR*** is an $N \times (N - 1)$ matrix, and ***PC*** is an $(N - 1) \times N$ matrix. The contributions of each pair of points to the matrices are determined by the equations (10) and (11). Intuitively, the greater value of an element implies higher probability that the corresponding segment is to be used in obtaining a shortest path/tree. Therefore, one can construct an optimal tree under statistical sense by finding a tree such that the sum of probabilities of all segments in the tree is maximized.

3. Algorithm

Based on the probabilistic model of RST, we describe an algorithm for Steiner tree construction as follows.

Probabilistic Analysis Algorithm:

Step 1: *Given **P** = {$p_1, p_2, \ldots, p_N$}, construct its grid graph, and compute the row number r(i) and column number c(i) for p_i, i = 1, 2, …, N;*

Step 2: *Compute the horizontal and vertical physical-lengths, i.e., H(k) and V(k) for k = 1, 2, …, N – 1;*

Step 3: *Compute F(m, n) and G(n, m) for m = 1, 2, …, N, and n = 0, 1, …, N – 1;*

Step 4: *Obtain the probability matrices, **PR** and **PC**, using equations (10) and (11) for all N· (N – 1)/2 pairs of points;*

Step 5: *Normalize all elements of **PR** and **PC** by setting*

$$\mathbf{PR}(i,j) \leftarrow \mathbf{PR}(i,j) / H(j), \quad 1 \le i \le N,\ 1 \le j \le N-1,$$

and

$$\mathbf{PC}(i,j) \leftarrow \mathbf{PC}(i,j) / V(i), \quad 1 \le i \le N-1,\ 1 \le j \le N;$$

Step 6: *Construct a tree **T** by selecting the segments one by one in the decreasing order of their corresponding probabilities in **PR** and **PC**, and performing the following three operations during the selection:*

- *(i) Ignore the current segment if selecting it would leads to a loop which contradicts the tree definition;*
- *(ii) Delete all redundant segments once all points in **P** have been connected by the tree;*
- *(iii) Calculate the total wire length of the obtained Steiner tree after the selection is completed.*

Most of the above algorithm is self-explanatory. In particular, the matrix normalization in Step 5 is necessary since the segments with same probability need to be treated differently, depending on their physical lengths. The shorter segment is selected first in the tree construction so that the total wire length can be minimized. If there are points with same *X*-coordinate, i.e., $H(j) = 0$, which implies that no horizontal segments in the *j*-th column are required, then we delete the *j*-th column of $\boldsymbol{PR}$ (instead of dividing it by $H(j)$ as shown in Step 5). Similarly, if some points have same *Y*-coordiate, i.e., $V(i) = 0$, meaning that no vertical segments in the *i*-th row are required, the *i*-th row of $\boldsymbol{PC}$ can be deleted.

In this algorithm, the most expensive computation occurs in Step 4 which takes $o(N^4)$ time. We have the following theorem without proof:

Theorem 2: The time complexity of the probabilistic analysis algorithm for Steiner trees is $o(N^4)$, where N is the number of given points.

4. Experiments and discussions

We implemented the proposed algorithm and carried out experiments with different number of points for Steiner tree construction. The results indicate that the obtained Steiner trees are better than those by the previous method, or very close to the optimal solutions. Several of our test examples are shown in Figures 4 through 6. While the optimal RST is unknown in general, especially, for the big problems with large value of *N*, the effectiveness of our approach can still be evidenced by inspection of these small-size problems. Particularly, for the case of Figure 6 which was taken from [5], the result due to the algorithm from [5] is shown in Figure 6(a) with the total wire length of 32, compared to the length of only 30 by our algorithm as shown in Figure 6(b).

In the following, we use Figure 6 to demonstrate the procedure of our algorithm presented in Section 3. From Steps 1 and 2, we can obtain four vectors which denote the row number, column number, horizontal physical-length and vertical physical-length. They are respectively: $\boldsymbol{r}$ = [2 6 5 1 3 4], $\boldsymbol{c}$ = [1 2 3 4 5 6], $\boldsymbol{H}$ = [2 3 1 7 1], and $\boldsymbol{V}$ = [2 2 1 1 5]. The values of $F(m, n)$ and $G(n, m)$ have been shown in Tables 1 and 2. After performing Steps 4 and 5, we have the probability matrices:

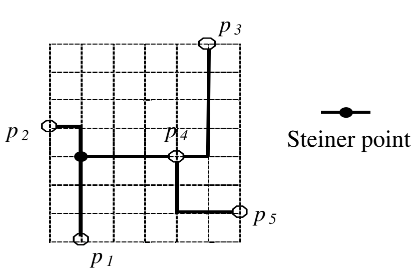


Figure 4. An example with $N = 5$

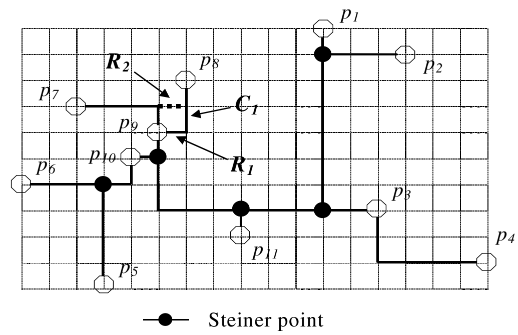
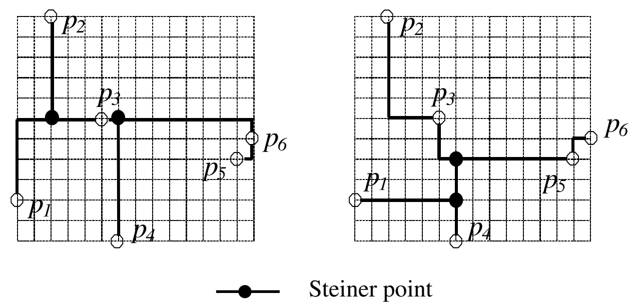


Figure 5. An example with $N = 11$
(note that replacing R_1 and C_1 with R_2 can lead to a better result)



(a) total wire length: 32 (b) total wire length: 30

Figure 6. Comparison of the results by (a) algorithm from [5] and (b) our algorithm based on probabilistic analysis

$$PR = \begin{bmatrix} 0.125 & 0.183 & 1.236 & 0.105 & 0.100 \\ 1.432 & 0.590 & 1.374 & 0.139 & 0.248 \\ 0.469 & 0.391 & 1.786 & 0.388 & 1.038 \\ 0.224 & 0.283 & 1.712 & 0.301 & 3.031 \\ 0.150 & 0.568 & 2.245 & 0.174 & 0.517 \\ 0.100 & 0.651 & 0.648 & 0.036 & 0.067 \end{bmatrix}$$

$$PC = \begin{bmatrix} 0.125 & 0.149 & 0.344 & 1.515 & 0.317 & 0.050 \\ 0.943 & 0.540 & 0.636 & 1.026 & 0.681 & 0.174 \\ 0.948 & 1.231 & 1.788 & 1.610 & 2.038 & 1.386 \\ 0.500 & 1.643 & 2.964 & 1.426 & 0.883 & 0.583 \\ 0.040 & 0.570 & 0.261 & 0.080 & 0.037 & 0.013 \end{bmatrix}$$

Finally, Step 6 gives the Steiner tree $\boldsymbol{T}$ = {R(2, 1), R(2,2), R(2, 3), R(3, 3), R(3, 4), R(4, 5), R(5, 2), C(1, 4), C(2, 4), C(3, 3), C(3, 5), C(4, 3), C(5, 2)}, as shown in Figure 6(b) which turns out to be an optimal RST. We claim that while the proposed algorithm produces promising results, it is generally not optimal. In Figure 5, for instance, a better solution could be found by replacing the segments R_1 and C_1 with the segment R_2 (shown as dotted line).

The main characteristics of our approach are two-fold. First, it is simple to implement and applicable to Steiner trees with any given set of points. Second, it can be extended easily to the more general Steiner problems with the obstacles or blockages (within their grid graphs) where any segment is prohibited [8], or with the routing congested regions where the segments are discouraged [9]. This can be done by properly assigning additional weights to the related elements of the probability matrices, so that the segments falling into these restricted areas would be less likely to be selected in the tree construction.

5. Conclusions

We have described an approach to Steiner tree problems based on probabilistic analysis. The best solution under statistical sense has been obtained for any given set of points. The results are better than those by the previous technique or very close to the optima. Further work includes comprehensive study and test on our model, so as to obtain the average performance of the algorithm on a large number of benchmarks. We also would like to extend the probabilistic model to applications for timing-driven interconnect optimization.

6. References

[1] F. K. Hwang, "An o(N log N) Algorithm for Rectilinear Minimal Spanning Trees," *Journal of the ACM*, vol. 26, pp. 177-182, 1979.

[2] M. R. Gray and D. S. Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete," *SIAM J. Appl. Math*, vol. 32, pp. 826-834, 1977.

[3] M. Hanan, "On Steiner's Problem with Rectilinear Distance," *SIAM J. Appl. Math*, pp. 255-265, 1966.

[4] J. P. Cohoon, D. S. Richards, and J. S. Salowe, "An Optimal Steiner Tree Algorithms for a Net whose Terminals Lie on the Perimeter of a Rectangle," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 4, pp. 398-407, April 1990.

[5] R. Condamoor and I. G. Tollis, "A New Heuristic for Rectilinear Steiner Trees," in *Proceedings of International Symposium on Circuits and Systems*, 1990, pp. 1676-1679.

[6] F. K. Hwang, "On Steiner Minimal Trees with Rectilinear Distance," *SIAM J. Appl. Math*, pp. 104-114, 1976.

[7] F. K Hwang and D. S. Richards, "Steiner Tree Problems," *Networks 22*, pp. 55-89, 1992.

[8] C. J. Alpert, et al, "Steiner Tree Optimization for Buffers, Blockages, and Bays," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 4, pp. 556-562, April 2001.

[9] E. Bozorgzadeh, R. Kastner, and M. Sarrafzadeh, "Creating and Exploiting Flexibility in Steiner Trees," in *Proceedings of 38th IEEE/ACM Design Automation Conference*, June 2001, pp. -.

[10] J. Cong, L. He, C. K. Koh, and P. H. Madden, "Performance Optimization of VLSI Interconnect Layout," *INTEGRATION, the VLSI Journal,* 21, pp. 1-94, 1996.