Minimum Steiner Tree Approximation using Binary Image Skeltons

Dionysios Kountanis Vatsal Gandhi Wasim ElHajj
Anupama Raju

{kountan,vsgandhi, welhajj, araju}@cs.wmich.edu
Department of Computer Science
Western Michigan University
MI-40096

Abstract

The problem of finding a Minimum Steiner Tree (MRST) for a set of points P on a rectilinear space has been proven to be NP-Complete. A binary image is constructed on the rectilinear space, which includes all the points in P. The skeleton of the binary image is used to develop a polynomial approximation algorithm (AMRST) for the MRST. The skeleton defines a tree with vertices $V = P \cup S$, where S is the set of Steiner points and a set of edges E that connect the vertices in V. Each edge $e = (v_i, v_j) \in E$ has a cost $c(e) = |x_i - x_j| + |y_i - y_j|$ where (x_i, y_i) and (x_j, y_j) are the coordinates of the vertices v_i and v_j respectively. Then, the cost of the AMRST is: $c(AMRST) = \sum_{e \in E} c(e)$ The cost of $c(AMRST)$ is compared with the cost of $c(MST)$ of the Minimum Spanning Tree (MST) of the set P. The maximum difference between the $c(MST)$ and $c(AMRST)$ is well-known, therefore, the proximity of AMRST to MRST can be measured.

Keywords: Minimum Steiner tree, minimum spanning tree, sketleton, binary image, rectilinear space

1 Introduction

Historically the research on Steiner points and Steiner trees started when Fermat posed the problem: Find a point in the plane that minimizes the summation of the distances from three given points. Steiner generalized the problem for n points. Because of its many applications, the problem enjoys great attention during the recent years.

A variation of the problem on the rectilinear space and on the grid (rectilinear with equidistant vertical and horizontal lines) is used in several phases of Computer Aided Design (CAD) for Very large Scale Integration (VLSI). Such phases include global routing, estimation of wiring length, etc. Other areas of application include Robotics, Network Communication Systems, Civil Engineering, etc.

The Steiner problem is defined as follows: Given a set of points P on the grid junctions, find a set S of Steiner points such that the minimum spanning tree for $P \cup S$ has minimum rectilinear cost. The rectilinear distance (cost) between two grid points (x_1, y_1) and (x_2, y_2) is given by $|x_1 - x_2| + |y_1 - y_2|$. The difference between a minimum spanning tree (MST) on the set P and the corresponding minimum Steiner tree (MRST) on the set $P \cup S$ is that MRST could contain additional grid points than merely the given points.

Hanan [4] states a fundamental result on the MRST for the rectilinear space. For a fixed number of points P the Steiner points S are located at the intersections of vertical and horizontal lines passing through the points in P.

Another fundamental result about Steiner points presented by Gilbert et.al [3] which states that for $|P| = n$ points there are at most $(n-2)$ Steiner points on the MRST. Figure 1 below illustrates a MST and a MRST for a set of 4 points. Notice that there are 2 Steiner points $(4 - 2 = 2)$ on the MRST and each Steiner point has degree equal to 3.

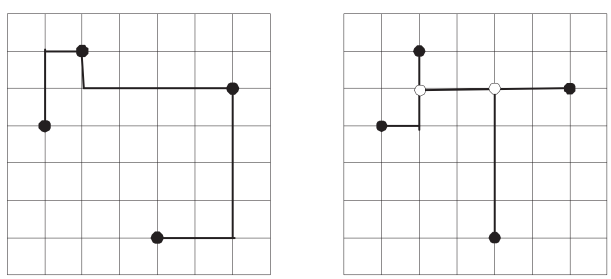

Figure 1: MST and MRST for 4 points

The MST problem is polynomial [5]. On the other hand, Garey and Johnson [2] proved that the MRST problem is NP-Complete, i.e., there is no polynomial algorithm that can possibly solve the MRST problem. This is true even if the space is restricted to be rectilinear.

Given that the MRST problem is NP-Complete, the research was directed towards finding good polynomial approximation algorithms for the MRST. A good algorithm finds Steiner trees with cost close to the cost of the MRST.

A simple approximation to the MRST is the MST which can be computed in polynomial time. Hwang [6] proved that for a fixed number of points on the rectilinear space the costs of the trees satisfy

$$\frac{cost(MST)}{cost(MRST)} \leq \frac{3}{2}$$

Hwang et.al [6] developed a $O(n \log n)$ approximation algorithm that produces rectilinear Steiner trees (RST) with good average performance. Kahng et.al [9] developed an iterated 1-Steiner tree (I1S) method to find MRST approximations. Basically the method works as follows: Compute MST for the given points P. Find a Hanan vertex H with the best improvement in the cost. Add H to P i.e., consider the new set $H \cup P$. Repeat the process. The best improvement point H is determined by the test:

$$MST(P \cup H) < MST(P)$$

The algorithm has order of complextity $O(n^4 \log n)$. Griffith et.al [7] presented a I1S approach with an $O(n^3)$ algorithm. Winter et.al [13] conducted a survey of Steiner tree applications on the Networks. The survey includes Steiner tree problems on the rectilinear space and on probabilistic networks as well as generalized Steiner trees. Ho et.al [8] developed approximation algorithms for the MRST based on the MST. Their approach, attempts to obtain maximum overlap of L-shape layouts and also separability of subtrees in the MST.

Kountanis et.al [10] considered the Steiner tree problem for Hypercubes. The problem remains NP-Complete. An $O(n^3)$ algorithm which approximates the MRST was designed, based on a Divide and Conquer strategy. The algorithm recursively computes local centers of gravity and constructs local approximation Steiner trees. These local trees are connected recursively to finally construct a global Steiner tree for the Hypercube. In [10] the authors used the concept of envelope to develop a process for approximating MRST. The process has $O(n^2 \log n)$ complexity and on the average the resulting tree has a 12.83% improved cost over the MST.

In this paper, a different approach is used based on image processing. The points on the grid are used to derive a binary image. Then, a modified skeleton algorithm operates on the image. The skeleton, by definition, is a stick like figure that illustrates the structure of the image. Figure 3 shows the skeletons of the images illustrated in Figure 2. Note, the illustrations are in the continuous space. On the discrete grid space the skeletons have the same structure.

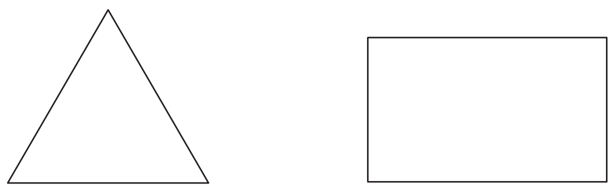

Figure 2: Two images, a triangle and a rectangle

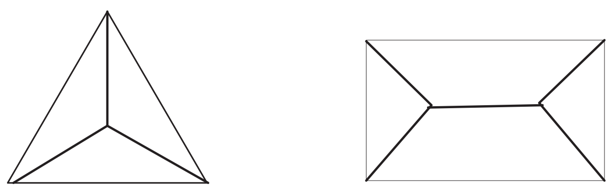

Figure 3: Two images, a triangle and a rectangle

In order to compute the Steiner tree, this paper uses a process of repeated erosion [1] of borderline points (pixels) by preserving the continuity of the image. A modification was introduced in order to avoid the erosion of points on the original set P.

2 Steiner Algorithm

In this section an approximation algorithm for the MRST is developed based on the skeleton of a binary image. A binary image is represented on the grid by binary numbers; 0 (white) 1 (dark). A pixel is a square defined in the grid. Each pixel is represented by the coordinates of its left-top corner.

Given a set of points P on the grid, the binary image is constructed as follows:

1. Compute the MST for the set P.

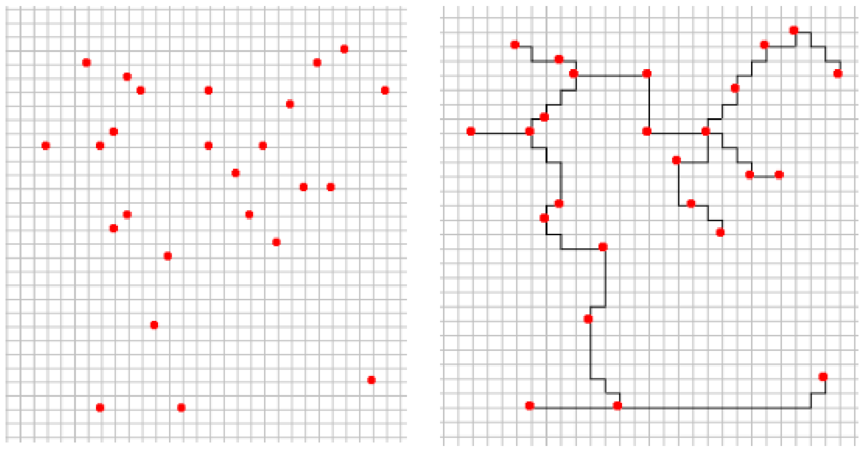

Figure 4: The grid and points P (left) and the corresponding MST (right)

2. Trace the grid from left to right, top to bottom. For every point of the grid that belongs to the set P form a rectangle of pixels from this point to its neighbor to the right. Note, the rectangles include the pixels that correspond to these two points.

3. Set to 1 the values if the remaining pixels which belong to a rectangle constructed in step 2.

4. Set to 0 the values of the remaining pixels. Figure 4 illustrates the set P, the MST for P and the images produced by the above process. The border pixels of the image are defined to be the ones with value 1 and at least a neighbor pixel with the value 0. The neighborhood is defined as follows: Let (x_i, y_i) a pixel in the image . The neighbors of (x_i, y_i) are $\{(x_i + 1, y_i)$, $(x_i - 1, y_i)$, $(x_i, y_i + 1)$, $(x_i, y_i - 1)\}$. The origin (0,0) of the image is the left top corner.

5. Apply the skeleton algorithm on the resulting image. The algorithm erodes (chips away) pixels with values of 1. The process repeats itself with the provisions; do not introduce discontinuity among the value 1 pixels and do not erode pixels corresponding to points in P. The result is a RST.

The Steiner algorithm based on the above description is presented below.

Input: The set $P = \{(x_i, y_i)$ where $1 \leq i \leq n$ are the given points$\}$ and the set $G = \{(x_i, y_i, x_j, y_j)$ are the set of edges of MST$\}$

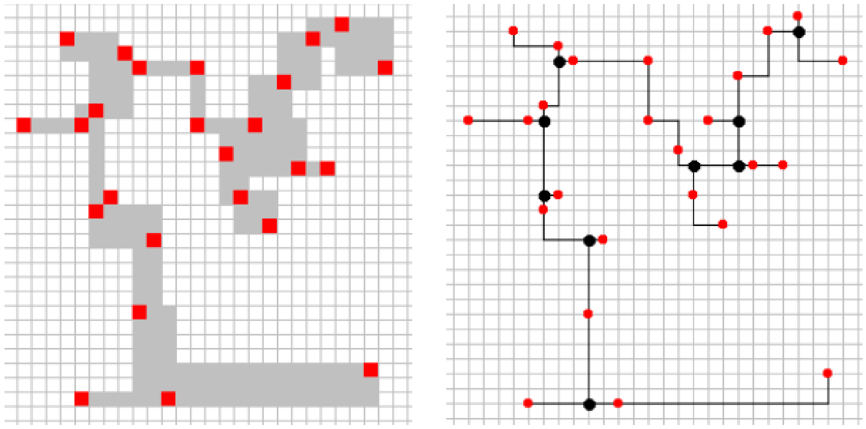

Figure 5: Image (left) and the corresponding Steiner tree (right)

Output: Steiner Tree

Step 1: Let D be $n \times n$ matrix which is used to store the binary image as pixels. For each set of edges $(x_i, y_i, x_j, y_j) \in G$ of MST, set the pixels on from (x_i, y_i) to (x_j, y_j) i.e., $x_i \to x_j$ and $y_i \to y_j$.

Step 2: For each pixel P_1 in D which is set
If $P_1 \in P$ i.e, Pixel is on of the point, then
Do not set it off
Else if P_1 is the end point pixel or is a connecting pixel or is a part of 2-pixel wide horizontal/vertical line then
Do not set it off
Else (for all other cases)
Set the pixel off
End For

3 Complexity of the Algorithm

The Algorithm consists of 2 steps:

Step 1 forms the image from MST. Since all the edges are to be considered, this step will be executed $(n-1)$ times.

Step 2 applies the modified skeletonization algorithm. It checks if the pixel which is a node/point of MST is not peeled off. Also, it checks if the pixel to be peeled off is not a connecting pixel or a end point. Since the above steps are performed for each and every pixel, it is executed $n \times n$ times. So,the overall time complexity of the algorithm is $O(n^2)$.

Table 1: Performance Comparison of MST and Steiner Trees

Points	MST	Steiner	Improvement
10	107.54	103.54	3.719546
15	122.3	117.7	3.761243
20	146.52	139.23	4.97543
25	166.78	156.98	5.876004
30	192.5	181	5.974026
35	198.45	186.42	6.06198
40	202	189.3	6.287129
45	239.32	224.13	6.34715
50	242	225.5	6.818182
55	251.3	233.93	6.912057
60	259.7	240.83	7.266076

4 Implementation

The Steiner algorithm was implemented in Java. The grid was set to size 50×50. The number of points in P was set to various values $n \in \{10, 15, 20, \ldots, 60\}$. For each n a multiple set of points, randomly generated, was set in the grid and the program was executed. The multiplicity of sets for each n was considered in order to reduce the influence of the point distributions on the results.

Table 1 shows the average costs for the Minimum Spanning Tree (MST), the Steiner Tree (RST) produced by the Steiner algorithm and the percentage improvements obtained on the cost of RST over the cost of MST. Figure 6 shows a pictorial representation of the data in Table 1.

It was observed from the output data that a uniform distribution of the points in the grid, for every n, resulted in the best approximation. It seems, by observation and not by proof, that the less uniform the distributions are, the worser approximations are achieved.

5 Conclusion

The idea of using image processing to compute approximations to the MRST is novice. The algorithm has very low complexity and produces

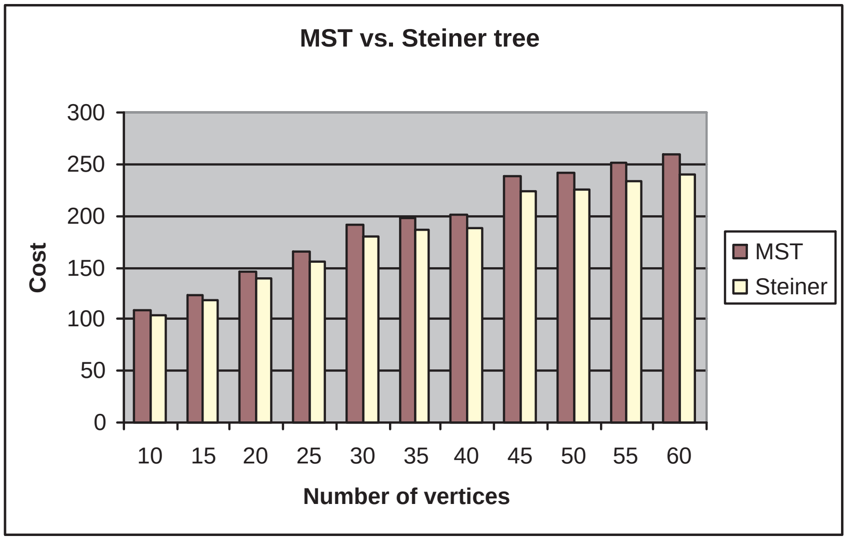

Figure 6: Histogram

good results. However, the representation of the points P by an image in the grid is not unique. There is a possibility that other image representations might be better that the one used. Using morphological operations on the image in order to find the skeleton is another possibility.

References

[1] E. Dougherty and C. Giardina. *Mathematical Methods for Artificial Intelligence and Autonomous Systems*. Prentice Hall, 1998.

[2] M. Garey and D. S. Johnson. The rectilinear steiner problem is np-complete. *SIAM J. Applied Math*, 32, 1977.

[3] E. N. Gilber and H. N. Pollak. Steiner minimal trees. *SIAM J. Applied Math*, 16, 1968.

[4] M. Hanan. On steiner's problem with rectilinear distance. *SIAM J. Applied Math*, 14, 1966.

[5] E. Horowitz and Sahni. *Fundamental of Computer Algorithms*. Computer Science Press, 1998.

[6] R. Hwang. On steiner minimal trees with rectilinear distance. *SIAM J. Applied Math*, 30, 1976.

[7] J. Salowe J. Griffith, G. Robins and T. Zhang. Closing the graph: Near-optimal steiner trees in polynomial time. *SIAM Journal of Discrete Mathematics*, 1994.

[8] G. Yijayam J. Ho and C. K. Wong. A new approach to the rectilinear steiner tree problem. In *26th ACM/IEEE Design Automation Conference*, 1989.

[9] A. B. Kahng and G. Robins. A new class of steiner tree heuristics with good performance: the iterated 1-steiner approach. In *IEEE International Conference on Computer Aided Design, Santa Clara*, 1990.

[10] D. Kountanis and K. Kokkinos. A balanced approach to the rectilinear steiner tree problem. *Congressus Numerantium*, 108, 1995.

[11] D. Kountanis and N. Liolios. Optimization of steiner nodes and trees on hypercube architectures. In *1st International Conference on Computer Communications and Networks*, 1992.

[12] D. Richards R. Hwang and P. Winter. The steiner tree problem. *Annals of Discrete Math, North Holland*, 53, 1992.

[13] P. Winter. Steiner problem in networks: A survey. *Networks*, 17, 1987.