J. Basic. Appl. Sci. Res., 3(2s)342-344, 2013
© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A Heuristic Approach to Construction Rectilinear Steiner Trees

Vahid Khosravinejad[1], Alireza Bagheri[2]

[1]Departement of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2]Department of Computer Engineering and Information Technology Amirkabir University of Technology, Tehran, Iran

Received: June 10 2013
Accepted: July 10 2013

ABSTRACT

The Steiner tree problem involves finding a minimal tree containing certain points and, if needed, a number of extra points for minimizing its length. The present paper investigated a Steiner tree with rectilinear edges called rectilinear Steiner tree. Rectilinear Steiner trees constitute the main part of routing in electronic integrated circuits. If all vertices in a rectilinear Steiner tree are leaves, then it is called full rectilinear Steiner tree. This paper aimed to introduce an approach for Construction of such trees. The results indicated its good efficiency.

KEYWORDS: Steiner tree, Rectilinear steiner tree, Full steiner tree.

1. INTRODUCTION

With p point(s) in a plane, the tree connecting p points to a set of q points is called the Steiner Tree (ST). The p points are the terminals and the q points are the Steiner points. Moreover, if the edges are rectilinear to each other, then it is called rectilinear Steiner tree. A large body of research has recently focused on the Steiner tree problem thanks to its vast applications in industrial engineering problems and wiring of electronic integrated circuits both vertical and horizontal wires [1-5]. The rectilinear Steiner tree problem is NP-complete and there is no algorithm with polynomial time for it [6].

The rectilinear Steiner tree problem can be classified into three major categories: (1) maze routing algorithm, which presents an optimal solution for networks with two terminals [7], (2) Non-deterministic approaches, which based on ant Colony and Genetic algorithms. In order to Construct a rectilinear Steiner tree, Hu et al. [8] and Julstrom [9] used Ants Colony and Genetic algorithms, respectively, (3) Graph-based approaches, which first create a spanning graph out of input terminals and then convert it into a rectilinear Steiner tree. This approach is employed in a study by Shen et al. [4].

The rest of the paper is organized as follows. Section 2 presents a few basic definitions. Section 3 introduces the Construction of Full Steiner Trees (FSTs). Section 4 discusses the proposed algorithm. The optimization of the proposed algorithm and computational results are presented in Sections 5 and 6, and the final section is the research conclusion.

2. Basic Definitions

A full Steiner tree, f, for a set, t_k, of k terminals is the Steiner tree which exactly has k-2 Steiner points and each Steiner point has a degree of 3; also, the angle between edges around these points is 120 degree. Furthermore, every terminal in f is necessarily has a degree of 1.

FST is a tree every terminal of which is a leaf (degree of 1) [9]. Edges in a rectilinear Steiner tree are perpendicular.

3. Construction of FST

In this section, Zachariasen and Winter's [10] algorithm for creating FST is presented.

Let p and q denote two points in the Euclidean space, e_{pq} would be the point obtained by rotating p counterclockwise by 60 degrees around p. In a counterclockwise fashion, p, q and e_{pq} constitute the vertices of an equilateral triangle. The circle encompassing $pe_{pq}q$ is called the equilateral circle of p and q and is denoted by c_{pq}. Its circumcenter is o_{pq}. The equilateral triangle and its equilateral circle are shown in Figure 1. The point r is placed in a way that the re_{pq} segment intersects pq arch at s. The point s is the Steiner point for p, q and r terminals and builds a 120 degree angle with each of them. The re_{pq} segment is known as the Simpson line of a FST with r, p and q terminals.

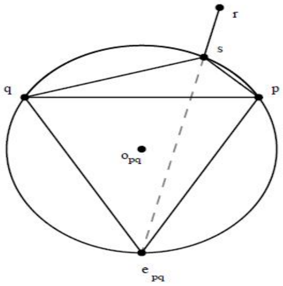


Figure 1. Method of obtaining Steiner point using Simpson line

---

***Corresponding Author:** Vahid Khosravinejad, Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran. E-Mail: V_khosravinejad@yahoo.com

Each FST could be created using the reversible feature of this principle. Figure 2 shows a sample FST for t_1 to t_6 terminals which are presented by bold lines.

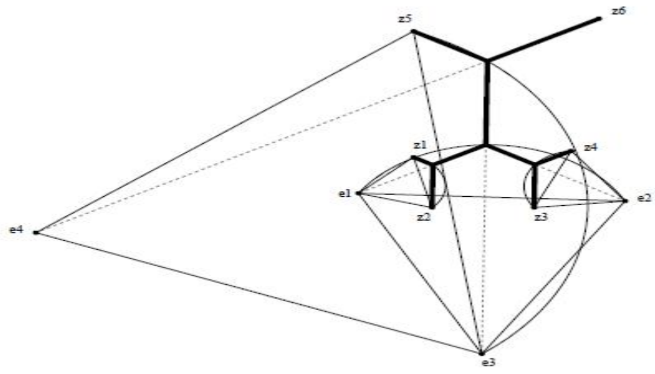


Figure 2. A sample FST

4. Proposed Algorithm for Construction Rectilinear FST

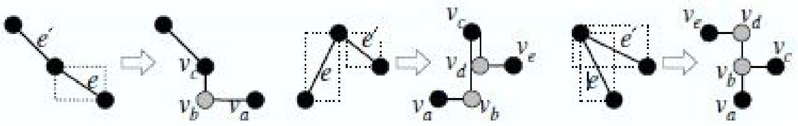
First, the input terminals are used to construct a FST based on the algorithm of the previous section. Next, each inclined edge is converted into horizontal and vertical edges and the result is a rectilinear FST. Suppose e and e' are two adjacent edges from FST. In order to render them rectilinear, the following three cases are considered:

Case 1. both edges are in opposite regions (Figure 3a). In this case, e is converted into two horizontal and vertical lines (Figure 3b). Here, there are two cases for turning them rectilinear, one of which is selected randomly.

Case 2. the edges' regions are adjacent (Figure 3c). In this case, e and e' are converted into horizontal and vertical edges. There are multiple cases for conversion. Here, the case in which edges become superimposed is selected (Figure 3d).

Case 3. both edges are in the same region (Figure 3e). In this case, Figure 3f is used.

For example, e and e' are transformed into (v_a, v_b) and (v_b, v_c), respectively. In this case, there are two possible cases for (v_c, v_e), one of which is selected randomly.



(a) (b) (c) (d) (e) (f)

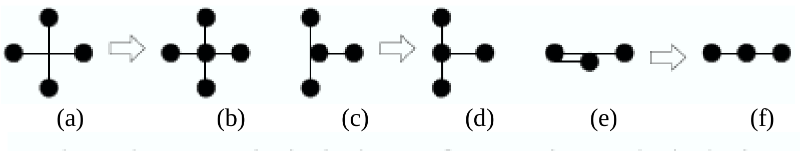
Figure 3. Cases for converting inclined edges into perpendicular

5. Optimization

Once the inclined edges of FST are converted into rectilinear, the following steps should be used for optimization of the rectilinear FST.

5.1.Removing superimposed edges

In the rectilinear FST from Section 4, if Figures 4 (a), (c), (e), (g), and (i) is the case, then it should be converted into Figures 4 (b), (d), (f), (h), and (j), respectively.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j)

Figure 4. Removing superimposed edges

5.2. Removing vertices redundancy

Definition 1. A redundant-vertex is a non-terminal with a degree of 2, and the both edges connected to it are parallel. In rectilinear FSTs, redundant vertices (Figure 5a) are removed and the edges connected to these vertices are connected to each other (Figure 5b).

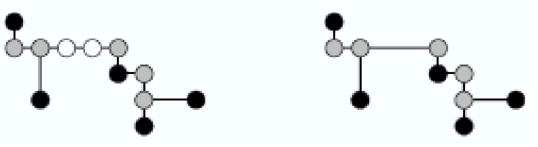


Figure 5. Removing redundant vertices

6. Computational Results

The proposed algorithm was implanted using C# programming language. Experiments were performed on examples from rc01-rc06 [11]. Table 1 compares a number of these results with those from [11]. As shown in Table 1, the proposed algorithm worked properly.

Table 1: The proposed algorithm in comparison with examples of [11]

Example	Lee [11]	Proposed rectilinear FST
Rc01	25980	25985
Rc02	42010	42013
Rc03	54390	54390
Rc04	59740	59742
Rc05	74650	74651
Rc06	81607	81607

7. Conclusion

The Steiner tree problem is widely used in physical designing of electronic integrated circuits all over the world. This paper proposed an algorithm for construction rectilinear FSTs and the results were compared with those from Lee et al. study. Finding rectilinear FSTs with obstruction is suggested for future studies.

REFERENCES

1. Z. Feng, Y. Hu, T. Jing, X. Hong, X. Hu, and G. Yan,” An O(nlogn) algorithm for obstacle- construction in the lambda-geometry plane," Proc. ISPD, pp.48-55, 2006.
2. Y. Hu, Z. Feng, T. Jing, X. Hong, Y. Yang, G. Yu, X. Hu, and G. Yan, “FORst: a 3-step heuristic for obstacle-avoiding rectilinear Steiner minimal tree construction," Journal of Information and Computational Science, pp. 107-116, 2004.
3. Y. Hu, T. Jing, X. Hong, Z. Feng, X. Hu, and G. Yan,”An-OARSMan: obstacle-avoiding routing tree Construction with good length performance," Proc. ASP-DAC, pp. 7-12,2005.
4. Z. Shen, C. Chu, and Y. Li, “Efficient rectilinear Steiner tree construction with rectilinear blockages," Proc. ICCD, pp.38-44, 2005.
5. Y. Shi, T. Jing, L. He, and Z. Feng, “CDC Tree: novel obstacle-avoiding routing tree construction based on current driven circuit model," Proc. ASP-DAC, pp. 630-635, 2006.
6. F. K. Hwang,”On Steiner minimal trees with rectilinear distance”. In Proc. of SIAM J.Appl. Math, 30:104–114, 1976.
7. C. Y. Lee, “An algorithm for connections and its application,"IRE Trans. on Electronic Computer, pp. 346-365, 1961.
8. B.A.Julstrom , " A scalable genetic algorithm for the rectilinear Steiner Problem " , IEEE 2002 .
9. F. K. Hwang, "On Steiner minimal trees with rectilinear distance," Proceedings SIAM Journal on Applied Mathematics, vol. 30, pp. 104-114, 1976.
10. P.Winter and M.Zachariasen , "Euclidean steiner minimum trees : an improved exact algorithm", Networks , Volume 30 ,Issue 3, Pages 149-166,1998.
11. Lee and Evangeline F. Y. Young, "Obstacle-avoiding rectilinear Steiner tree construction," in Proc. Int. Conf. Computer.-Aided Des., pp. 523-528,2008.