J. Basic. Appl. Sci. Res., 3(1s)611-613, 2013
© 2013, TextRoad Publication

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

A Heuristic Algorithm for Construction Euclidean Steiner Minimal Tree Inside Orthogonal Polygon

Alireza 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 2 2013
Accepted: July 6 2013

ABSTRACT

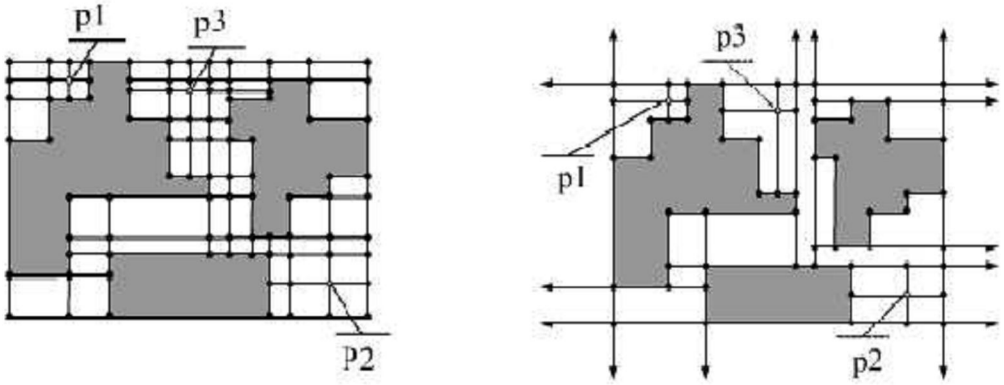
Steiner tree problem consists of finding the minimum tree that includes specific points and, if necessary, uses a number of auxiliary points to minimize the tree length. In this paper, a case study of Steiner tree in one whose vertices and edges are placed inside an orthogonal polygon. This problem is widely used in oil lines, highways and electronic integrated circuit design. This is an Np-hard problem and no polynomial-time algorithm has yet been discovered for it. We propose an algorithm to construct the Steiner tree using a graph and have reached optimal solutions in some cases, in this paper.
KEYWORDS: Euclidean steiner minimal tree, Steiner tree in graph, escape graph, track graph and orthogonal polygon.

1. INTRODUCTION

Steiner tree problem is widely used in scientific and commercial fields such as computer networks routing, electronic integrated circuits, electrical cabling and design of highways. Having n point set in plane, the tree that connects n points and a set of q points together is called Steiner. Steiner tree with minimal length in Euclidean Plane is called Euclidean Steiner Minimal tree. This problem is known as NP-hard [1] and no polynomial-time algorithm has been presented for it. In this heuristic algorithm, the problem has been transform to Steiner tree problem in graph, trying to achieve optimal or near-optimal answers. A lot of efforts have been done to solve the Steiner tree problem [2, 3]. In recent years, iterative algorithms such as genetic algorithm [4] and ant colony algorithm [5] have been proposed to solve Steiner tree problem. These methods do not guarantee an optimal response, but in most cases they will lead to reasonable approximate solutions. The rest of the paper is organized as follows. In section 2, two types of connected graphs are introduced. Section 3 presents the proposed algorithm. Computational results and conclusions are expressed in section 4 and 5, respectively.

2. Connected Graphs

Ganley et al. presented a strong connected graph which is known to escape graph and proved that all Steiner points in optimal solution exist on this graph (Figure 1-a) [6]. Wu et al. offered a connected graph with grid structure named track graph including vertical routes which are defined by obstacles and terminals (Figure 1-b) [7].



(b) (a)

Figure1. Escape graph structure and track graph for three terminals and three obstacles

The number of vertices and edges in an escape graph is equal to $O((1+n)^2)$, where L is the number of obstacles' the boundary points and n is the number of terminals. The number of vertices and edges in a track graph is (r^2), where r is maximum boundary edge of all obstacles. Track graph is not a strong connected graph. It means that in some cases, this graph does not include an optimal solution. However, this graph usually consists of optimal or near-optimal solutions can.

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

In large scales, the number of edges and vertices in escape graph is much more than track graph, therefore the escape graph is used in most algorithms.

3. Heuristic Algorithm

We offer our Heuristic algorithm in three steps. First, an escape graph is created in orthogonal polygons. In step 2, we add a set of points as Steiner points and edges to the graph in step 2. Steiner tree in graph is obtained in step 3.

Step1: Construction A Graph G

Suppose p is the set of vertices of the polygon and n is the terminals set, first we perform a horizontal scan on the entire polygon which restricts the terminals. In this scan, when we get each vertex $V \in (P \bigcup n)$, we will extend it vertically up and down along the inside of the polygon. Suppose the intersection of these two line segments with polygon boundary is v_1 and v_2. We add v, v_1 and v_2 to all vertices of the graph. If the new edge added to G has intersection with the previous edges of the graph, intersection points of this edge will be added to G as new vertices (Figure 2).

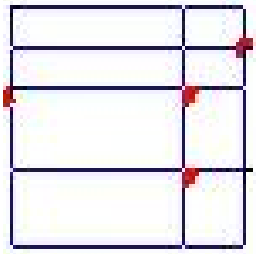


Figure 2. Construction a Graph G by scanning operation

Step2: Adding Steiner points and edges to the graph G

Each rectangle formed by scanning operation in Step 1 is triangulated by following method. Suppose one of the rectangles formed from previous step (figure 3) has t_1, t_2, t_3 and t_4 vertices. First, we draw an edge from t_1 to t_3 and t_2 to t_4. We name two resulted triangles $\Delta t_1t_2t_3$ and $\Delta\ t_2t_3t_4$, respectively. Angles of $\Delta t_1t_2t_3$ and $\Delta t_2t_3t_4$ triangles are less than 120°. Steiner points in each triangle are obtained using Torricelli method and called s_1 and s_2. Then we connect s_1 to t_1, t_2 and s_2 to t_4, t_3 and s_1 to s_2 with edges (Figure 3). Then we add point s_1 and s_2 and edges t_1s_1, t_2s_1, t_3s_2, t_4s_2 and s_1s_2 graph G vertices and edges.

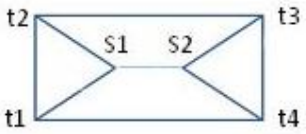


Figure 3. Steiner points and edges obtained from triangulation

Figure 4 is obtained after adding all Steiner points and edges.

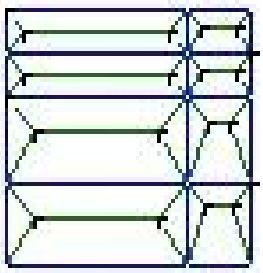


Figure 4. Steiner points and edges obtained from step 2

Step3: Using Dreyfuss and Wagner algorithm [8], Steiner tree which contains all the terminals in the graph is obtained.

4. Computational Results

We have implemented the proposed algorithm by c# programming language. We performed our experiments by examples of Soukup [9]. In Table 1, a number of our results are compared to optimum results. As it is shown in Table 1, the proposed algorithm has acceptable results. Figure 5 illustrates the resulting tree form implementation of the soukup problem 11 algorithm.

Table 1. Our algorithm compared with Soukup examples

Example number	Optimal result	Our proposed algorithm
EX.3	159.88	159.88
EX.5	164.83	165.70
EX.6	128.62	129.30
EX.7	220.49	220.94
EX.9	116.68	118.31
EX.12	172.22	173.59

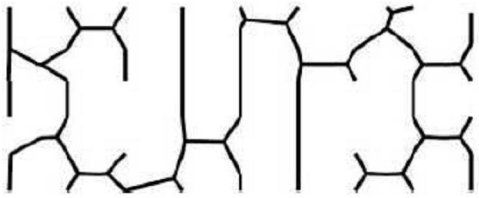


Figure 5. Implementation of proposed algorithm for Soukup Example 11

5. Conclusion

The Steiner tree problem is one of common scientific and commercial ones; which several researches are constantly performed on it; around the world. In this paper we have presented an algorithm that uses entry points to form a graph. Other points and edges are added to the algorithm to achieve optimal results.

REFERENCES

1. M. R. Garey, R.L.Graham, D.S. Johnson, “The complexity of computing Steiner minimal trees”, SIAM J. Appl. Math, 1977
2. Y. T. Tsai,CH. tang, and Y.Y.Chen, ”An Average Case Analysis of a Greedy Algorithm for the On-Line Steiner Tree Problem “, Computre Math .Applic.,Vol.31,No.11,pp.121-131,1996.
3. B. M. Waxman, “Routing of multipoint connections”, IEEE Journal on Selected Areas in Communications.Vol.6, No.9, pp.1611-1622, 1998.
4. s. ding, and N. Ishii,”an online Genetic Algorithm for Dynamic Steiner Tree Problem”, Proc,Symposium on Computational geometry.pp.337-343, 1995
5. A.Norollah , A.Jahanian , P.Adibi, M.Tashakori , " solving dynamic Steiner tree with ant colony " , Iran , computer society , 1383.
6. J.L.Ganley and J.P.Cohoon, " Routing a multi – terminal critical net: Steiner tree construction in the Presence of obstacles “, in Proc .OF IEEE ISCAS, London, UK: pp.113-116, 1994.
7. Y.F.Wu, P.Widmayer , M.D.F. Schlag and C.K. Wong ,"Rectilinear shortest tracks and minimum spanning trees in the presence of Rectilinear obstacles" ,IEEE Trans on Computers, 36(3) :pp.321-331,1997.
8. S.-E. Dreyfus, R.-A. Wagner, “The Steiner Problem in Graphs”, Networks, Vol.1, PP. 195-207, 1972.
9. J.Soukup and W.F.Chow “set of test problems for the minimum length connection networks”, IEEE, 1986.