J. Basic. Appl. Sci. Res., 3(2s)345-348, 2013
© 2013, TextRoad Publication

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

Euclidean Steiner Minimal Tree Inside Simple Polygon with Presence Obstacles

Vahid Khosravinejad[1], Alireza Bagheri[2]

[1]Department 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

Steiner tree problem leads to solutions in several scientific and business contexts, including computer networks routing and electronic integrated circuits. Computing fields of this problem has become an important research topic in computational geometry. Considering the number of points in the Euclidean plane, called terminal points, a minimum spanning tree is obtained which connects these points. A series of other points (Steiner points) are added to the tree, which makes it shorter in length. The resulting tree is called Euclidean Steiner minimal tree. It is considered as an NP-hard problem. Considering a simple polygon P with m vertices and n terminals, in which you are trying to find a Euclidean Steiner tree that is connected to all n terminals existing inside p. In this paper we propose a solution for several terminals in a simple polygonal in presence of obstacles.

KEYWORDS: Steiner tree, Euclidean steiner minimal tree, Delaunay triangulation, simple polygon.

1. INTRODUCTION

Steiner problem can be used in scientific and commercial fields such as computer networks routing and integrated electronic circuits, oil distribution and transport network. Computational fields of this problem make it an important research topic in computational geometry. Considering a few points in Euclidean plane, the shortest Path connecting these points lead to the attainment of a tree is called Euclidean Steiner minimal tree. Euclidean Steiner minimal tree is considered as an NP-hard problem [1]. In this paper, we obtain a Euclidean Steiner minimal tree (ESMT), which has n terminals in a simple polygon with m vertices and several obstacles. It has been proven that ESMTs connecting 4 terminals [4] proposed an a algorithm for the ESMT Lee et al.together have high-quality solutions for cases with no obstacles [2,3]. subset of each of these terminals within a simple polygon with O(K log k), where K = m + n. They determined 4, 3, 2 have provided an exact algorithm inside a simple polygon. Steiner tree is obtained from any subsets' vertices. Winter et al. 4 terminal in a simple polygon [6], for the three terminals in a simple polygon in time O(K) [5], an exact algorithm for heuristic algorithm for over 4 terminal [7]. The proposed algorithm in this paper can solve the Steiner tree in a simple polygon with some obstacles. Finally, we compare our results with the data and results presented in [8]. The article is organized as follows; the proposed algorithm is introduced in section 2. In section 3, the proposed algorithms' steps are described in detail. The computational results are presented in section 4. The final section brings other sections in a conclusion.

2. The Proposed Algorithm Steps

A simple polygon with m vertices, n terminals and a number of obstacles is presented as an input, as shown in Figure 1.

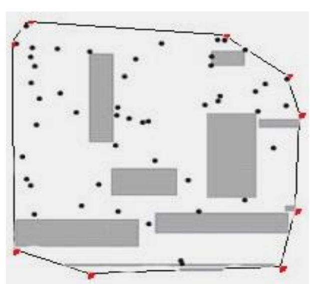

Figure 1. A simple polygon m vertices, n terminals and a number of obstacles

Our proposed algorithm has three steps.

Step1. First according to [9] we form Delaunay triangulation using sets of terminals and obstacles' corners. The geometrical shape is shown in Fig 2. Then we remove the triangles' edges within the obstacles. It is shown in Figure 3.

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

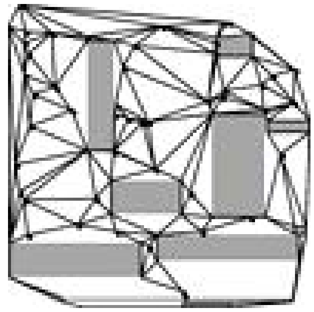

Figure 3. Obstacles preventing triangulation

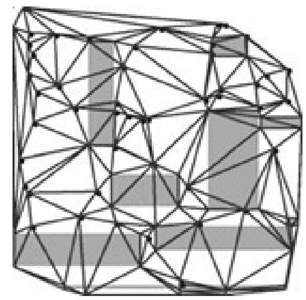

Figure 2. Delaunay triangulation

After removing the edges placed within the obstacles, a triangulation is achieved, called obstacle-avoiding Constrained Delaunay triangulation (OACDT).

Step2. The minimum spanning tree is obtained from OACDT using Prim's algorithm [10] (Figure 4). Then, we remove all the non-terminal leaves from OACDT. The resulting tree is called T_f (figure 5). This step is fully explained in Section 3.2.

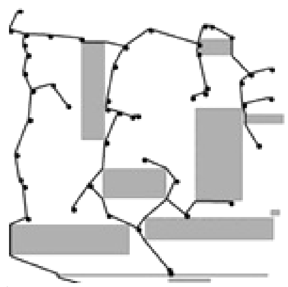

Figure 5. T_f Tree

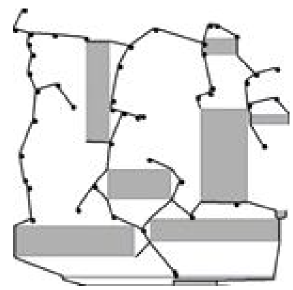

Figure 4. Minimum Spanning Tree

Step3. T_f tree is triangulated and the Steiner point is obtained in each triangle. The resulting tree ESMT tree is described in Section 3-3.

3. Details of the Proposed Algorithm

3.1 Step1: OACDT Construction

First, regardless of input terminals set, vertices of the polygon and the set of obstacles' vertices that make up the corners of obstacles, we construct the Delaunay triangulation (DT). This triangulation is performed during O(nlogn), where n is the set of terminal, polygon and obstacles' vertices [9]. The edges of each triangle with obstacles are evaluated using 3.1.1 algorithm to find out whether they intersect each other or not. The edges will be removed if they have intersection. This evaluation requires $C \times n_e \times n_o$ time unit, where C indicates the number of obstacles' edges, n_e demonstrates the number of edges and n_e and n_o are the number of DT edges and number of obstacles, respectively.

3.1.1 Intersection Algorithm

Suppose AB as triangle side, and CD as obstacle border. If spot route ABD is left-handed and ABC is right-handed (vice versa), side AB and border CD will intersect each other. We can calculate the determinant of points' coordinates to find out whether their route is right handed or left handed. Statement 1 is an example of such determinant which is considered for three points A, B and D.

$$\begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_D & y_D & 1 \end{vmatrix}$$

Statement 1

If the determinant calculation result is greater than zero, three-point route is right-handed and this route is left-handed, if the result is smaller than zero.

3.2 Step2: T_f Construction

The purpose of this step is to construct a tree on OACDT that connects all terminals.
corners on the obstacles. All leaves in T_f are **1(T_f) :** T_f is a tree which connects all terminals, OACDT **Definition** terminals. First, the Minimum Spanning Tree is generated by Prim algorithm which connects all terminals and obstacles' corners on ACDT. Then, the minimum spanning tree is surveyed using depth first search (DFS) and all the non-terminal T_f.leaves are removed Terminals. This resulting tree is called

3.3 Step3: ESMT Construction

In this section, ESMT of T_f made in section 3-2 is built and Steiner points regarding to the triangulation of T_f , described in two steps, is added.

Step1. T_f vertices set are triangulated as follows; for all T_f vertices of degree greater than one, initially, we order all edges connected to that vertex angularly. Then for instance, we connect either edge of this ordering located beside each other. All these three edges form a triangle. We consider a number equivalent to the total size of the two edges of T_f, involved in triangle forming. For each triangle, the vertex based on which the triangle is formed (joint vertex between two edges of T_f) involved in triangle forming, is called triangle central vertex. According to figure 6, each edge (ab here) will be involved in up to 4 triangles.

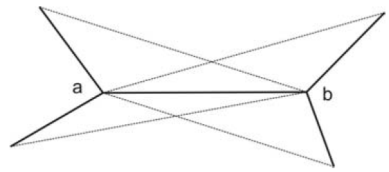

Figure 6. Maximum number of triangulation which would happen for each edge

Step2. In each abc triangle with angles less than 120°, there is a point “S” that the length of lines connecting it to a, b and c vertices is less than the length of each two sides of the triangle [11]. “S” is called the Steiner point. This step of the algorithm is repeated equal to the number of triangles generated in the previous step. Each time, the triangle that the largest number is attributed to it, and has not been investigated for existence of Steiner Point is chosen. After finding each Steiner point in this set, triangles will be updated by the method set forth in section 3.3.1.

3.3.1 Updating Triangles

Suppose triangle abc is the one selected from the previous step whose sides are ab and ac from T_f. If the vertices a, b and c are connected to $a_1,a_2,\dots,a_m$ and $b_1,b_2,\dots,b_n$ and $c_1,c_2,\dots,c_p$, a triangle with sides ac and ab will be updated as follows (figure 7) .

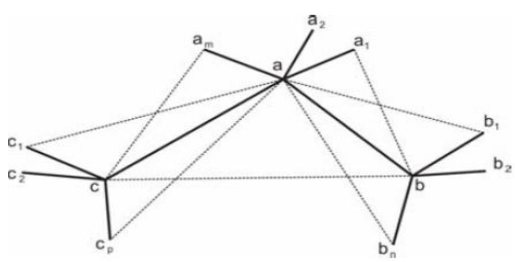

Figure 7. Possible triangles around two edges of T_f

Suppose that point α is the Steiner points in triangle abc. Apart from the triangle abc, those triangles with ab edge upgrades in this way; if their central vertex is a and b, this edge will be replaced by aα and bα, respectively. The same method is applied to the triangles with ac edges. The resulting tree after these steps is called ESMT.

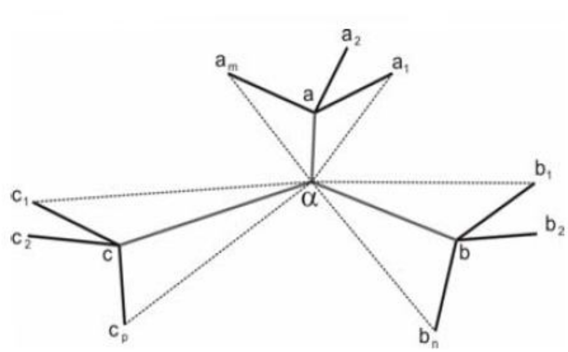

Figure 8. Updated triangles from adding Steiner point in figure 7

4. Computational Results

We implemented the proposed algorithm in C# programming language and performed our experiments with examples of Soukup [8]. We considered a convex polygon around all the terminals and assumed that there is no terminal in the obstacles. Then we compared some of our results with optimum results in table 1. The presented algorithm provides good results as shown in Table 1.

Table 1. Proposed algorithm compared with Soukup examples

Example number	Optimal result	Our proposed algorithm
EX.1	166.44	166.44
EX.2E	220.53	220.53
EX.5	164.83	164.90
EX.7	220.49	220.80
EX.13	103.96	103.96
EX.18	104.21	105.21
EX.20	222.95	224.95

5. Conclusion

This paper presents an algorithm capable of solving Steiner tree problem inside a simple polygon with some obstacles in the Euclidean plane. Computational results of stated algorithm presented. Resulting algorithm is simple in terms of implementation and leads to good 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. J.E. Beasley, "A heuristic for the Euclidean and rectilinear Steiner tree problem", European J. Operational Research. 58 (1992) 284–292.
3. J. Macgregor Smith, D.T. Lee, J.S. Liebman, "AnO(nlogn) heuristic for the Steiner minimal tree problem on the Euclidean metric", Networks 11 (1981) 23–29.
4. E.Papadopoulou, D.T. Lee, "A new approach for the geodesic Voronoi diagram of points in a simple polygon and other restricted polygonal domains", Algorithmica 20 (1998) 319–352.
5. P. Winter, "Euclidean Steiner minimum trees for 3 terminals in a simple polygon", Proceedings of the Seventh Canadian Conference on Computational Geometry, Univ. Laval, Quebec, Canada, 1995 ,pp.247–255.
6. M. Zachariasen, P. Winter, "Obstacle-avoiding Euclidean Steiner trees in the plane: an exact algorithm", in: M.T. Goodrich, C.C. McGeoch (Eds.),Algorithm Engineering and Experimentation, Lecture Notes in Computer Science, Vol. 1619, Springer, Berlin, 1999, pp. 282–295.
7. P. Winter, M. Zachariasen, J.Nielsen, "Short trees in polygons", Discrete Applied Mathemati cs, Volume 118, Issues 1-2, pages 55-72, 2002.
8. J. Soukup, W.F. Chow, "set of test problems for the minimum length connection networks" ACM SIGMAP Bulletin, 1973.
9. S.Fortune, "A sweepline algorithm for Voronoi diagrams.Algorithmica", 2(2):pp.153-174, 1987.
10. R.C. Prim, "Shortest connection networks and some generalizations", Bell System Tech. J.36 (1957):pp.1389-1401.
11. A.O.Ivanov and A.A.Tuzhilim , " Minimal Networks : The Steiner Problem and its Generalizations " , CRC Press , 1994.