J. Basic. Appl. Sci. Res., 3(2s)309-311, 2013
© 2013, TextRoad Publication

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

A Heuristic Algorithm for Construction the Steiner Tree Inside Simple polygon in the Presence of Obstacles

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

On the whole, the Steiner tree problem is finding a minimum tree which consists of specific points and if need be, uses a number of auxiliary points in order to minimize the length of the tree. In this paper, the Steiner tree is investigated in a case in which the vertices and edges of the tree are within the simple polygon P. It is also assumed that there are a number of obstacles in form of simple polygons within the polygon P with which the edges of the Steiner tree should not intersect. This way of expressing the problem indicates the limitations which are to be encountered in the real world in instances such as manufacture of integrated circuits as circuit elements or in different kinds of routing where there are geographical obstacles. In this paper, efforts have been made to achieve optimum responses by turning this problem into a graph problem.
KEYWORDS: Steiner tree, Euclidean steiner minimal tree, Steiner tree in graph, escape graph.

1. INTRODUCTION

The Steiner tree problem has applications in scientific and commercial fields such as computer network routing, design of electronic integrated circuits, post networks, and oil well networks, in a way that macrocells and geographical obstacles are considered as obstacles in the routing phase. This problem is an NP-hard one [1]. Having the set P of N points within a simple polygon in the presence of a number of obstacles, a tree connecting the points of P and a set of points Q is called the Steiner tree (ST). The points of P are called terminals and the points of Q are called Steiner points. The Steiner tree with minimum length on the Euclidean plane is called the Euclidean Steiner Minimal Tree (ESMT). The problem aims at finding a Steiner tree in the Euclidean space for n terminals without any collision with any of the obstacles. Numerous algorithms for solving the Steiner tree problem have been presented so far owing to its importance. Precise algorithms which normally use dynamic programming methods and branch and cut [2, 3, 4] and have exponential complexity in the worst scenario, are not suitable for use in practical applications. For this reason, several approximation algorithms have been presented for solving the Steiner tree problem [5, 6, 7]. The rest of the paper is organized as follows: In section 2, two types of connection graphs are introduced. In section 3, the proposed algorithm is presented. In section 4, the calculation results are presented and in section 5 the conclusion is expressed.

2. Connection graphs

Ganley et al. presented a strong connection graph which is known as escape graph. They proved that all of the Steiner points in the optimum solution exist in this graph (Figure 1-a) [8]. Wu et al. presented a connection graph called track graph with a structure resembling that of a net which comprises rectilinear paths defined by obstacles and terminals (Figure 1-b) [9].

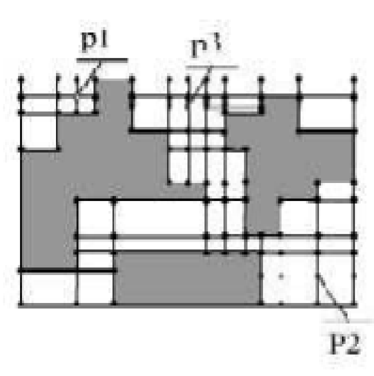


Figure 1-b. A track graph

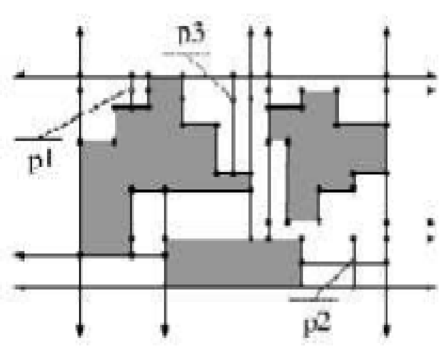


Figure 1-a. An escape graph

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

The number of vertices and edges in an escape graph is O(l+n), where L is the number of border points of obstacles and n is the number of terminals. The number of vertices and edges in a track graph is $O(r^2)$, where r is the maximum number of border edges of all obstacles.

2.1. Reducing the track graph

Assume that C is a vertex of a convex non-terminal corner in an obstacle in Figure 2 and V_1 and V_2 are the vertices adjacent to C. Suppose that V_1 and V_2 are both vertices of a concave corner and also assume that V_3 and V_4 are the vertices adjacent to V_1 and V_2 (except C). If we have the edges V_3V_5 and V_4V_5 in a track graph, then the vertices C, V_2, V_1, and also the edges V_3V_1, V_2V_4, V_1C, and CV_2 are omissible from the graph (Figure 2).

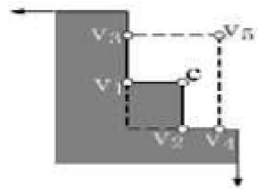
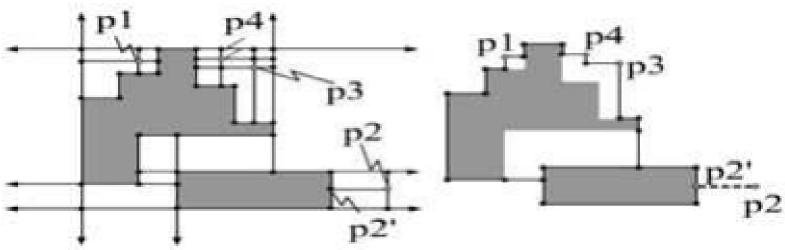


Figure 2. Omitting unnecessary edges and vertices in a path graph

At the end of this omission, we will have a number of first order terminals. Experience shows that by hypothetically omitting these terminals and projection them onto the adjacent vertex (which will be considered as terminal from now on), a number of terminals will be projected onto one point. Once calculations are done, these edges will be added to the resulting graph. Figure (3-a) shows the structure of a track graph in the case when there are 4 terminals and 2 obstacles and Figure (3-b) depicts the structure of the reduced graph.



(a) (b)

Figure 3. Structure of a track graph in the reduced state

3. The Heuristic Algorithm

The heuristic algorithm is presented in three steps. In the first step, graph G is constructed within a polygon. In the second step, the Steiner tree is obtained on the graph and in the third step, the Steiner point in the obtained tree is optimized.

Step 1: Constructing graph G

Suppose that P is the set of polygon vertices and O is the set of obstacle vertices and n is the set of terminals. At first, we run a horizontal sweep on the entire polygon restricting the terminals and obstacles. In this sweep, once we get to any vertex $V \in (P \cup O \cup n)$, we extend it vertically upward and downward within the polygon (if possible). Assume that the intersection point of these two lines with the perimeter of the polygon or obstacles are the points V_1 and V_2. We add the vertices V, V_1, and V_2 to the set of vertices of graph G and also the edges VV_1 and VV_2 to the set of its edges. If the new edges added to G intersect with its previous edges, the intersection points of these edges are added to G as new vertices (Figure 4). This graph (G) is called a polygon graph. We then reduce the obtained graph using section 2.1.

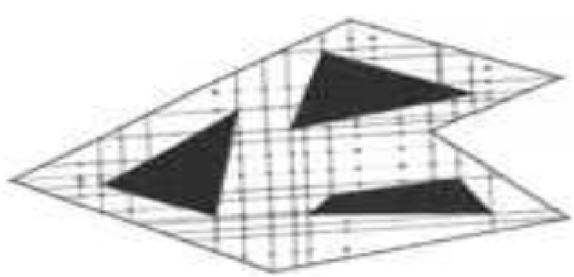


Figure 4. Graph G

Step 2: The Steiner tree on graph G including all input terminals is obtained using the algorithm of Beasley et al. [10].
Step 3: In the tree obtained in section 2, the Steiner point for each vertex of the terminals and Steiner points which has an order equal to or greater than 2 is obtained using Simpson's method [11], provided that it has an angle of less than 120 degrees with its two adjacent vertices. We then investigate whether the overall length of the edges of these three vertices connected to the Steiner point is less than that of the three edges of the tree. If it is so, the Steiner point is added to the tree and the tree edges are omitted and the edges connected from these three vertices to the Steiner point are added to the tree.

4. Calculation Results

The proposed algorithm was implemented using C# programming language. The experiments were conducted using examples from Soukup [12]. 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 shows the tree yielded by running the proposed algorithm.

Table 1. Proposed algorithm compared to Soukup's examples

Example Number	Optimum result	Our proposed algorithm
EX.2A	207.77	208.44
EX.4	127.41	128.25
EX.10	164.28	165.12
EX.11	382.80	383.05
EX.18	104.21	104.80
EX.20	222.95	223.93

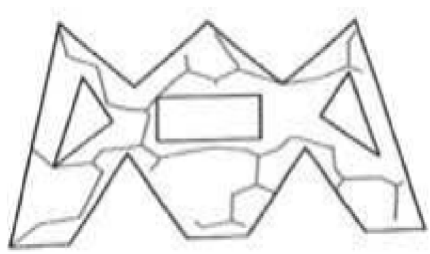


Figure 5. Result of running the proposed algorithm within a polygon

5. Conclusion

The Steiner tree problem is one of the most widely used scientific and commercial problems. Various researches are permanently conducted on it throughout the world. In this paper, an algorithm was presented which could solve the Steiner tree problem within a simple polygon in the presence of obstacles. The calculation results of the algorithm suggest that the algorithm yields responses at least 1.01 the optimum ones.

REFERENCES

1. M. R. Garey, R.L. Graham, D.S. Johnson, "The complexity of computing Steiner minimal trees", SIAM J. Appl. Math., 1977.
2. S. E. Dreyfuss, and R. A. Wagner, "The Steiner Problem in Graphs", Networks,Vol.l, pp.195-207,1971.
3. S. Chopra, E. R.Gorres, and M. R. Gorres and M. R. Rao,"Solving the Steiner tree problem on a graph using branch and cut", ORSA Journal on Computing, Vol.4, pp.320335, 1992.
4. S. Milan, and Mirko, "An Exact Algorithm for Steiner Tree Problem on Graphs", International Journal of Computers, Communications & Control,Vol.I,No.l,pp.41-46,2006.
5. H. Takahashi, and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs", Mathematica Japonica, Vol.24, No.6, pp.573-577, 1980.
6. V. J. Rayward-smith, and A. Clare, "On Finding Steiner Vertices", Networks, Vol.24, No.6, pp.573-577, 1980.
7. L. Kou, G. Markowsky, and l. Breman, "A Fast Algorithm for Steiner Trees", Acta Informatica, Vol. 15, pp.141-145, 1981.
8. 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.113116,1994.
9. Y. F. Wu, P. Widmayer, M. D. F. Schlag and C. K. Wong, "Rectilinear Shortest paths and minimum spanning trees in the presence of Rectilinear obstacles", IEEE Trans on Computers,36(3) :pp.321-331,1997.
10. Lucena, J.E. Beasley, "A branch and cut algorithm for the Steiner problem in graphs", Networks, Vol.31,pp.39-59,1998.
11. G. Ausiello, P. Crescenzi, G. Gambosi, V. kann, A. Marchetti Spaccamela and M. protasi, Complexity and Approximation, Springler – Verlag, 1999.
12. J. Soukup and W. F. Chow, "Set of test problems for the minimum length connection networks", IEEE, 1986.