International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 3, June 2017, pp. 1262~1267
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i3.pp1262-1267

An Improved Augmented Line Segment based Algorithm for the Generation of Rectilinear Steiner Minimum Tree

Vani V[1], G. R. Prasad[2]
[1]Department of Information Science Engineering, Bangalore Institute of Engineering, Bangalore, India.
[2]Department of Computer Science and Engineering, BMSCE, Bangalore, India.

Article Info

Article history:

Received Jan 15, 2017
Revised Apr 15, 2017
Accepted Apr 29, 2017

Keyword:

Global routing
Rectilinear minimum spanning tree
Rectilinear steiner minimum tree
VLSI design

ABSTRACT

An improved Augmented Line Segment Based (ALSB) algorithm for the construction of Rectilinear Steiner Minimum Tree using augmented line segments is proposed. The proposed algorithm works by incrementally increasing the length of line segments drawn from all the points in four directions. The edges are incrementally added to the tree when two line segments intersect. The reduction in cost is obtained by postponing the addition of the edge into the tree when both the edges (upper and lower L-shaped layouts) are of same length or there is no overlap. The improvement is focused on reduction of the cost of the tree and the number of times the line segments are augmented. Instead of increasing the length of line segments by 1, the line segments length are doubled each time until they cross the intersection point between them. The proposed algorithm reduces the wire length and produces good reduction in the number of times the line segments are incremented. Rectilinear Steiner Minimum Tree has the main application in the global routing phase of VLSI design. The proposed improved ALSB algorithm efficiently constructs RSMT for the set of circuits in IBM benchmark.

Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Vani V,
Department of Information Science and Engineering,
Bangalore Institute of Technology,
K R Road, V V Puram, Bangalore, Karnatake-560004, India.
Email: vanisrin@gmail.com

1. INTRODUCTION

Rectilinear Steiner Minimum Tree (RSMT) is the tree that connects the given set of points in a rectilinear fashion using only horizontal and vertical line segments. Few additional points called as Steiner points are added to the tree. Steiner points are added to connect the edges in rectilinear manner and to reduce the total cost of the tree. The rectilinear distance between two points $p_1=\{x_1, y_1\}$ and $p_2=\{x_2, y_2\}$ is given by $|x_1-x_2| + |y_1-y_2|$. The main application of RSMT is in the global routing phase of the VLSI design. During global routing, the wiring channels (points) that have been placed at particular position using placement algorithm will be connected in rectilinear manner.

Figure 1 illustrates the RSMT constructed over the points P_1 to P_8 respectively. It can be identified that all the edges of the RSMT are the part of Hanan Grid. An edge can be either degenerate or non-degenerate. An edge is degenerate, if the two points it connects are in the same x-axis or y-axis. The non-degenerate edge for connecting two points is obtained from the enclosing rectangle and can be either the upper L-Shaped layout or the lower L-shaped layout. The cost or the total length of the RSMT can be reduced by carefully selecting an appropriate layout for each edge that overlaps with the layout of the other edges. For example in the Figure 1, the lower L-shaped layouts are selected for connecting the two edges P_1-P_2 and P_2-P_6 as they result in overlap and hence cost reduction. Rectilinear Steiner Minimum Tree construction is one of the fundamental problems that have many applications in VLSI design. It can be used for wire load estimation, identifying routing congestion and interconnect delay in early stages of VLSI design.

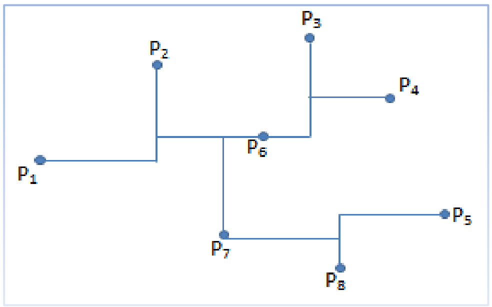

Figure 1. Rectilinear Steiner Minimum Tree

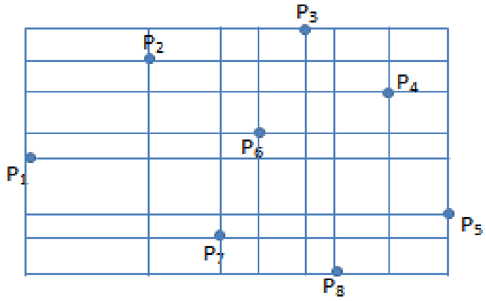

Figure 2. Hanan Grid

Rectilinear version of Steiner Tree was first studied by Hanan who gave exact solutions for the number of points (N) ≤ 5 and also proved that the Steiner points lie on the Hanan Grid. Hanan grid is obtained by inducing horizontal and vertical line segments through the given set of points as shown in figure 2. The points at the intersection of these line segments are called Hanan Points [1]. Many approximation algorithms exist for constructing RSMT as the problem of constructing RSMT is shown to be NP complete by Garey and Johnson [2]. The ratio of the length of rectilinear version of Minimum Spanning Tree (RMST) to that of RSMT is proved to be ≤ 3/2 by Hwang [3].

Khang and Robins proposed Iterated 1- Steiner (I1S) algorithm that iteratively adds a Steiner point that results in cost reduction and a Batched version (Batched Iterated 1-Steiner algorithm) where a group of Steiner points are added during each iteration [4]. Borah, Owens and Irwin presented an edge-based heuristic algorithm which initially constructs a Minimum Spanning Tree and transforms it into a RSMT by iteratively connecting a point to the enclosing rectangular layout of the visible edge in the MST [5]. Zhou proposed a Rectilinear Spanning graph (RSG) algorithm [6] which applies Borah et al edge based heuristic algorithm[5] on Spanning Tree constructed from the Zhou at al spanning graph algorithm which was constructed by connecting each point to the nearest point in eight octal regions [7].

Griffith, Jeff, et al proposed a variant of BI1S using dynamic MST update scheme where a point is connected to the nearest points in eight octants and the longest edge is removed in the formed loop[8]. Khang, Mandoiu and Zelikovsky proposed a batched version of greedy triple contraction algorithm[9] called Batched Greedy Algorithm where RSMT is constructed by iteratively adding a batch of triples (optimal full Steiner tree for a set of 3 points with all the points in the leaves position) [10]. Wong, Yiu-Chung and Chu proposed a Fast Look-Up Table based algorithm with a pre-computed table for constructing RSMT for N≤9. For N>9, a net breaking algorithm is iteratively used until N<9 and the pre-computed table can be used [11]. RSMT was also constructed by connecting the trees that have been constructed for the computed clusters of given points [12]. The existing algorithms for the construction of RSMT have been extensively surveyed [13].

2. RESEARCH METHOD

The proposed algorithm works by drawing incremental four line segments through each and every point. The edges are iteratively added when two line segments intersect. For each of the edge, two L-shaped layouts can be identified. The L-shaped layout which has an overlap with other edges should be cleverly selected to reduce the overall cost of the RSMT. If a decision in selecting a layout cannot me made or if both are of same length, the process of adding the edge to the RSMT will be delayed until proper decision cannot be made. The enhancement to Augment Line Segment Based (ALSB) Algorithm is carried out by doubling the size of line segments in each iteration unless they cross the intersection point [14]. If they cross intersection point, the length of the line segments will be reduced back to the previous value and augmenting starts will value of 1. The procedure carried out is as follows:-

1. Identify the boundary- Boundary is computed by identifying the minimum and maximum x and y values. The length of the line segments are incremented until they touch the boundary or until the RSMT is constructed.

2. Augment the line segments- The length of the line segments are doubled each time until it crosses the intersection point of any two line segments else the length of all the line segments will be set to the previous values and again starts augmenting with the step_size of one.
3. Construct RSMT- RSMT will be constructed by incrementally adding edges when two line segments intersect and if that edge does not form a loop in the partially constructed RSMT. Adding an edge requires selecting one of the two L-Shaped layout which results in cost reduction. If both the edges are of same length, then both edges will be marked as temporary edges until a decision can be made.

Finally when no more edges can be added and if temporary edges exist then the L-shaped layouts are checked for overlap with the constructed RSMT and the corresponding Layout will be added.

2.1. Algorithm

The improved ALSB algorithm takes as an input a set of points and computes the RSMT along with its cost or total length.

```
n---number of points
Current_length=0
step_size=0
while (num_edges<n-1)
begin
        for i=1 to n ---do in parallel
                previous_length =Current_length of line segments
                if (step_size==0)
                        Current_length= previous_length+1
                        step_size=1;
                else
                        Current_length=previous_length+2*step_size
        for i=1 to n ---do in parallel
        begin
                if (two line segments intersect && doesn't form loop in constructed RSMT)
                begin
                        compute length of upper L-shaped layout (length1) and lower L-shaped
                                layout (length2)
                        if (length1==length2)
                        begin
                                if (degenerate edge)
                                        add edge to RSMT
                                else
                                        indicate both as temporary
                        end
                        else
                        begin
                                add the shorter edge to RSMT
                                if (length reduction because of temporary edge)
                                       make that edge permanent and add to RSMT
                                        Remove the other contemporary temporary edge
                        end
                end
        end
        if (line segments cross the intersection point)
                step_size=0
                Current_length=previous_length
end

if (there are temporary edges that need to be added to the tree)
begin
        Compute the length of upper L-shaped layout (length1) and lower L-shaped layout (length2)
        if(length1==length2)
```

randomly add any one L-shaped layout
else
add the L-shaped layout with reduced length
end

The RSMT length reduction is obtained by further checking for overlap when the temporary edges are to be added to the constructed tree rather than randomly selecting any one as in ALSB algorithm. The number of line segment increment is reduced by doubling the step_size until the line segments cross the intersection point.

3. RESULTS AND ANALYSIS

The proposed algorithm has been implemented in C. Figure 3 shows that the improved ALSB algorithm shows good improvement over the ALSB algorithm with respect to number of times the line segments are augmented on various set of points that were randomly generated. When the points are less and sparsely located, good reduction can be identified. Table 1 demonstrates the cost reduction obtained by the improved ALSB algorithm.

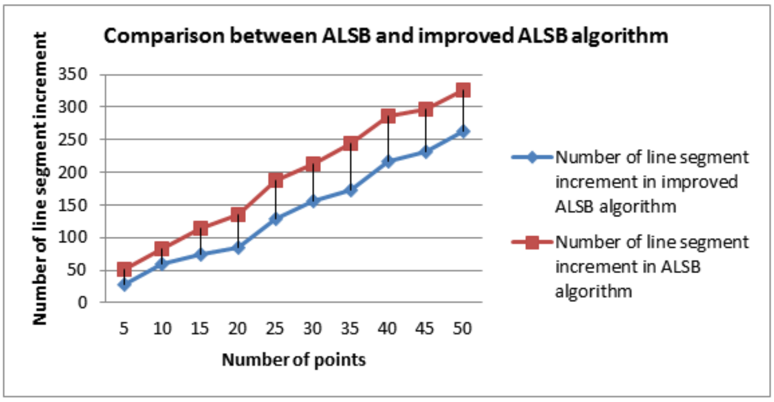

Figure 3. Comparison between ALSB and improved ALSB algorithm

Table 1. Comparison of the cost of RMST, RSMT using ALSB, RSMT using improved ALSB algorithm

Number of points	Total length of Rectilinear Minimum Spanning Tree	Total length of RSMT using ALSB algorithm	Total length of RSMT using improved ALSB algorithm
10	832	791	786
20	1344	1305	1294
30	1916	1787	1773
40	2196	1975	1958
50	2497	2336	2321
100	3257	3156	3132
200	4682	4376	4345
300	5849	5198	5173
400	6679	6056	6026
500	7407	6594	6559
600	8054	7130	7105

The algorithm was tested on IBM ISPD08 benchmark for global routing where the placement was carried out by Dragon 1.0 [15]. The benchmark information is listed in the Table 2 with the details of the number of nets in each circuit. It can also be identified that most of the nets in each circuit have degree < 10.

Table 3 shows the results obtained by the application of improved ALSB algorithm on IBM benchmarks. RSMT was effectively constructed for all the 10 circuits and the maximum RSMT length computed for a net in each circuit is as shown in Table 3.

Table 2: IBM benchmark for global routing

Circuit	# of nets	# of nets with degree <=10	# of nets with degree >10	Average degree	Maximum degree
ibm01	11508	10897	611	3.847	42
ibm02	18430	17211	1219	4.052	134
ibm03	19735	18759	976	3.195	53
ibm04	26164	25679	485	3.424	46
ibm05	27778	24124	3654	4.480	17
ibm06	33355	31014	2341	3.726	35
ibm07	44395	41671	2724	3.702	25
ibm08	47495	44719	3226	4.133	75
ibm09	50394	47542	2852	3.728	39
ibm10	64228	59583	4645	4.188	41

Table 3: Application of improved ALSB algorithm on IBM ISPD08 benchmark

Circuit	# of nets	Total RSMT length of all nets	Max RSMT length of a net in the circuit
ibm01	11508	61967	118
ibm02	18430	172778	366
ibm03	19735	138681	176
ibm04	26164	167795	269
ibm05	27778	438022	329
ibm06	33355	300093	295
ibm07	44395	378053	209
ibm08	47495	428319	414
ibm09	50394	429567	264
ibm10	64228	597643	333

4. CONCLUSION

The proposed improved ALSB algorithm provides good improvement over the ALSB algorithm in terms of number of line segment increment and cost reduction. The algorithm was also efficiently tested on IBM ISPD08 benchmark as shown in table 3. Future efforts would be directed towards further cost reduction of the tree and to implement the above algorithm on FPGA to improve the performance.

REFERENCES

[1] Hanan, Maurice, "On Steiner's Problem with Rectilinear Distance", *SIAM Journal on Applied Mathematics*, 14.2 (1966): 255-265

[2] Garey, Michael R, David S. Johnson, "The Rectilinear Steiner tree Problem is NP-Complete*", SIAM Journal on Applied Mathematics*, 32.4 (1977): 826-834.

[3] Hwang, Frank K, "On Steiner Minimal trees with Rectilinear Distance", *SIAM Journal on Applied Mathematics*, 30.1 (1976): 104-114.

[4] Kahng, Andrew B., Gabriel Robins, "A New Class of Iterative Steiner tree Heuristics with Good Performance", *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on 11.7 (1992): 893-902.

[5] Borah, Manjit, Robert Michael Owens, Mary Jane Irwin, "An Edge-based Heuristic for Steiner Routing*", Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on 13.12 (1994): 1563-1568.

[6] Zhou, Hai, *"Efficient Steiner ree Construction based on Spanning Graphs",* Proceedings of the 2003 international symposium on Physical design. ACM, 2003.

[7] Zhou, Hai, Narendra Shenoy, William Nicholls, *"Efficient Minimum Spanning tree Construction without Delaunay Triangulation",* Proceedings of the 2001 Asia and South Pacific Design Automation Conference. ACM, 2001.

[8] Griffith, Jeff, et al, "Closing the gap: Near-Optimal Steiner trees in Polynomial Time", *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on 13.11 (1994): 1351-1365.

[9] Kahng, Andrew B., Ion I. Mandoiu, Alexander Z. Zelikovsky, *"Highly Scalable Algorithms for Rectilinear and Octilinear Steiner trees"*, Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific. IEEE, 2003.

[10] Zelikovsky, Alexander Z, "An 11/6-Approximation Algorithm for the Network Steiner Problem*", Algorithmica*, 9.5 (1993): 463-470.

[11] Wong, Yiu-Chung, Chris Chu, *“A Scalable and Accurate Rectilinear Steiner minimal tree Algorithm”,* VLSI Design, Automation and Test, 2008. VLSI-DAT 2008. IEEE International Symposium on. IEEE, 2008.
[12] Vani, V., G. R. Prasad, *“Algorithm for the Construction of Rectilinear Steiner Minimum tree by identifying the Clusters of Points”*, Information Communication and Embedded Systems (ICICES), 2014 International Conference on. IEEE, 2014.
[13] Vani, Prasad, “Performance Analysis of the Algorithms for the Construction of Rectilinear Steiner Minimum tree”, *IOSR Journal of VLSI and Signal Processing*, Volume 3, Issue 3(Oct 2013), 61-68.
[14] Vani, Prasad, *“Augmented Line Segment based Algorithm for Constructing Rectilinear Steiner Minimum tree”,* Proceedings of IEEE International Conference on Communication and Electronics Systems ,2016, in press.
[15] The ISPD98 Circuit Benchmark Suite.http://vlsicad.ucsd.edu/UCLAWeb/cheese/ispd98.html

BIOGRAPHIES OF AUTHORS

Vani V is an Assistant Professor in the Department of Information Science and Engineering, Bangalore Institute Of Technology, Bangalore. She received B.E degree in Computer Science and Engineering from Visveshwaraiah Technological University in 2003 and M.Tech degree in Computer Network Engineering from VTU in 2007. She is currently pursuing Ph.D. degree in VTU in the area of Reconfigurable Computing.

Prasad G R is an Associate Professor in Department of Computer Science & Engineering, BMSCE, Bangalore. He holds a Ph.D from National Institute of Technology, Karnataka, Surathkal, INDIA. He received his M.Tech degree in Computer Science & Engineering from Bangalore University in 1999 and B.E Degree in Computer Science & Engineering from Bangalore University in 1995. His research interests include Reconfigurable computing.