International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 7, Issue 1, January - February 2018 ISSN 2278-6856

Rectilinear Steiner Minimum Tree Formation using an Improved Augmented Line Segment Based Algorithm with Edge Reversal

Vani V[1] and G R Prasad[2]

[1]Department of Information Science Engineering, Bangalore Institute of Engineering, V.V.Puram, Bangalore-560004

[2]Department of Computer Science and Engineering, BMSCE, Basavanagudi, Bangalore-560019

Abstract

An improvement to Augmented Line Segment Based Algorithm is proposed incorporating Edge reversal to generate a Rectilinear Steiner Minimum Tree with an objective of tree length reduction. Rectilinear Steiner Minimum Tree connects the given collection of points by means of horizontal and vertical line segments, with the aid of additional points called Steiner Points. Steiner points are introduced to reduce the total length of the tree and to connect in rectilinear manner since the distance between two points is measured in rectilinear metric. The algorithm constructs a Rectilinear Steiner Minimum Tree by augmenting the line segments drawn in four directions from all the points and adding an edge to the tree when two line segments overlap. The reversal of the two L-shaped layouts or edges is done when an overlap count of either of one increases thereby reducing the cost or length of the final tree formed. Rectilinear Steiner Minimum Tree is widely used in global routing phase of VLSI design and wire length estimation.

Keywords: Rectilinear Steiner Minimum Tree, Global Routing, Rectilinear Minimum Spanning Tree, VLSI Design

1. INTRODUCTION

The routing phase in VLSI design takes the cells which have been assigned positions as input and connects the multi-terminal nets using wires that run in horizontal and vertical directions because of manufacturing constraint. Routing is carried out in two different phases. Global routing is the first phase and it identifies the wiring channels through which connections can be made. Detailed routing is the second stage during which the exact path will be fixed. Rectilinear Steiner Minimum Tree (RSMT) has a wide application in the global routing phase of VLSI design. RSMT is a tree with following properties :(i) It connects the given set of points without any loop. (ii)The points are connected in rectilinear fashion using only horizontal and vertical line segments.

Given a collection of points Q, RSMT is a tree constructed over Q U S with line segments restricted to horizontal and vertical lines and S is a collection of Steiner Points found at the intersection of two edges (figure 2a). Steiner points with degree ≥2 are added to S. The distance between two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is equal to $|x_1-x_2| + |y_1-y_2|$. Figure 1a and 1b shows the Minimum Spanning Tree and the RSMT constructed for the given set of points respectively.

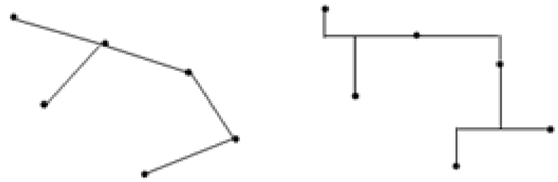

Figure 1. a) Minimum Spanning Tree **b)** RSMT

Hanan et al [1] provided the prime contribution by considering the rectilinear version of Steiner Tree, made available the precise results for n≤5 and also showed Hanan Grid includes all the probable Candidate Steiner points (CSp) (figure 2b). Garey et al [2] proved that the problem of generating RSMT is NP-Complete and Hwang et al [3] showed that the ratio of the cost of Rectilinear Minimum Spanning Tree (RMST) to the cost of RSMT is ≤ 3/2.

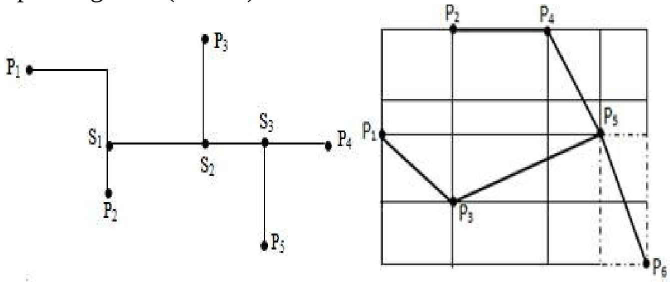

Figure 2. a) Rectilinear Minimum Tree **b).** Hanan Grid

Spanning Tree based algorithms construct the MST first using any spanning Tree algorithm and then convert it into RSMT. Ho et al [4] converted the MST by replacing each MST edge with an L or Z-shaped layouts that provided maximum overlap. Khang et al [5] introduced an Iterated Steiner Tree algorithm which iteratively adds a Steiner points and constructs the RSMT until no more improvement can be done to the RSMT. A batched version was also proposed to add a batch of Steiner points at a time. Borah et al [6] presented an edge-based heuristic algorithm that iteratively connects a point in the MST to the encompassing rectangular layout of nearby visible edge and then removes the lengthiest edge in the formed loop [9].

Chu et al [7] proposed a method of constructing RSMT on the basis of pre-computed table. A look up table was pre computed for n≤9. For n>9, net breaking algorithm was

International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 7, Issue 1, January - February 2018 **ISSN 2278-6856**

used to split the points recursively until n≤9 and the table can be applied. Chen et al [8] developed a Refined Single Trunk algorithm which first constructs a trunk which is a horizontal or vertical line that goes through the median of the given points. Later all the points are either connected to the trunk or the stem (a horizontal or vertical line that connects a point to the trunk) based on the shortest distance. Vani et al [9] proposed a clustering based RSMT algorithm which first divides the set of points to various clusters and then computes the RSMT for each cluster of points. Final RSMT is obtained by connecting these clusters.

Augmented line segment based algorithm [10] was presented where the length of four line segments from all the points were incrementally increased and edges were added to the tree when two line segments intersect. An improvement was provided by doubling the length of line segments until it goes further the intersection point. If so, the increment begins from one again [11]. The algorithms for the production of RSMT have been comprehensively reviewed in [12].

3 BASIC TERMINOLOGIES

- Minimum Spanning Tree - Given a group of points P, Minimum Spanning Tree (MST) is the tree with minimum length connecting all the points in P. Fig. 2b shows the MST constructed over the points p1 to p6.
- Rectilinear Minimum Spanning Tree (RMST) - MST with distance between two points measured in rectilinear metric and the edges restricted to horizontal and vertical line segments [9].
- Hanan Grid - Grid obtained by drawing horizontal and vertical lines through each and every point in P (figure 3). The points obtained at the juncture of these lines are called as Hanan points [1].
- L-Shaped Layout - For each non-degenerate edge (an edge is said to be non-degenerate if the edge end points do not fall on the identical vertical or horizontal line) two L-shaped layouts can be formed from the enfolding rectangle. In figure 3 the two L-Shaped layouts for the edge (P5, P6) are indicated using dotted lines [12].

2.RESEARCH METHOD

The proposed algorithm provides an improvement to Augmented Line Segment Based Algorithm [11] by introducing an L-shaped layout reversal of the previously selected layout based on the overlap count. Overlap count is computed by counting the number of points in the current edge that are already the part of other edge previously selected. If the overlap count of previously selected layout is less than the count of previously rejected layout, then the reversal takes place. The overlap count of previously rejected layout increases when a new edge overlaps with this edge. Hence, edge reversal plays a vital role in length reduction of the final RSMT.

2.1Algorithm

1 Augment the length of line segments drawn from each point in all the four directions.
2 When two line segments intersect and do not form a loop in the partially constructed RSMT, compute the overlap count of the two L-shaped layouts. Overlap count identifies the extent the L-shaped layout overlaps with partially constructed RSMT.
3 If both the L-shaped layouts have the same overlap count, then both are selected as temporary edges. Temporary edges will be made permanent only when it overlaps with another edge. Overlap is identified when the new edge is being laid on the same set of points of the previously selected temporary or permanent edge.
4 If either of the L-shaped layouts has more overlap count because of the overlap with other temporary edge, then that L-shaped layout with more overlap count is made permanent and the other layout is marked as rejected.

2.2 Illustration

Consider the set of points as shown in the figure 3a. The RSMT constructed by the Augmented Line Segment Based Algorithm (ALSB) [11] is shown in figure 3b.

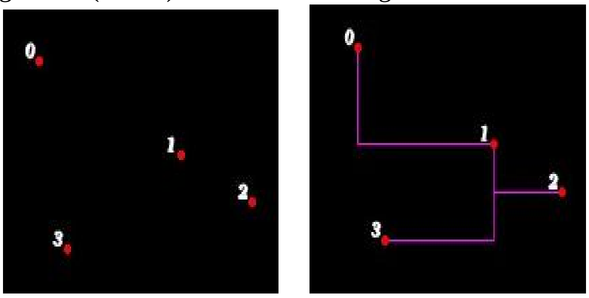

Figure 3. a) Given set of 4 points **b)** RSMT constructed using ALSB algorithm

The snapshots illustrating the Improved Augmented Line Segment Based algorithm with Edge Reversal is shown below. Figure 4 shows the first step of the algorithm wherein all the four line segments from all the points are augmented.

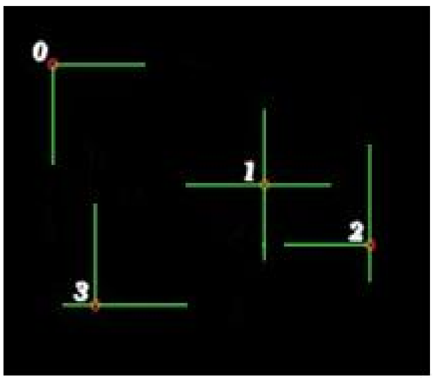

Figure 4. Augmentation of line segments from all the points in four directions

Figure 5 shows that when the line segments of points P_1 and P_2 intersect, both the L-shaped layouts are marked as temporary as they have zero overlap count (no overlap). But when the line segments of points P_1 and P_3 intersect, the lower L-shaped layout is selected and marked as permanent as it has more overlap count (overlap with the temporary edge connecting points P_1 and P_2). The alternate L-shaped layouts of edge (P_1, P2) and edge (P_1, P3) are marked rejected as shown in figure 6. Pink and brown color is used to represent the permanent and rejected edges respectively.

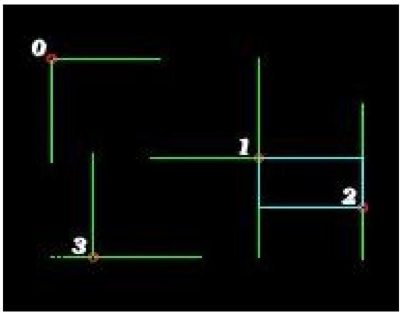

Figure 5. L-shaped layout of P_1 and P_2 marked as temporary

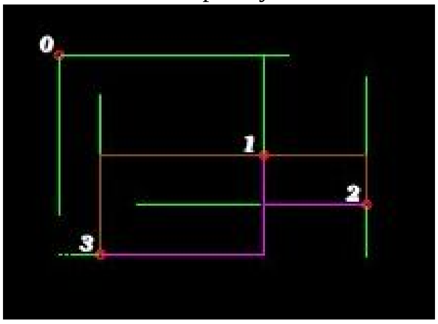

Figure 6. Layouts of edge (P_1, P2) and edge (P_1, P3) marked as selected and alternative layouts marked as rejected

Figure 7 shows edge reversal. Here the lower L-shaped layout of edge (P_1, P_3) which was selected or marked as permanent previously is rejected and the other contemporary upper L-shaped layout is selected as it has more overlap count when the edge (P_0, P_1) is connected. The final RSMT constructed by the Improved Augmented Line Segment Based algorithm with Edge Reversal is as shown in figure 8. It can be very clearly observed that the length of this RSMT is less compared to the one generated by ALSB algorithm (figure 3b).

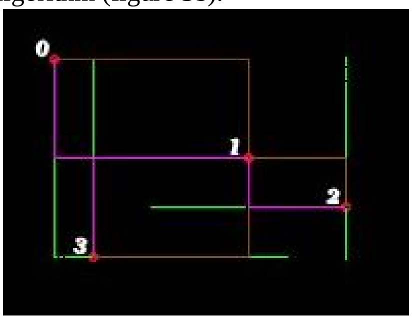

Figure 7. Edge reversal showing the reversal of layouts of edge (P_1, P_3)

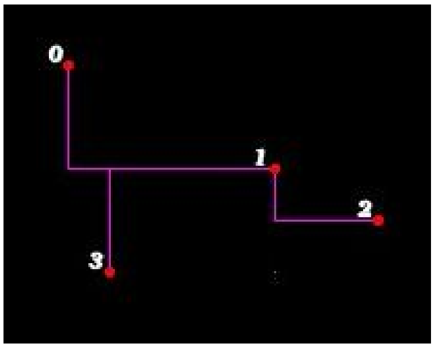

Figure 8. RSMT constructed by Improved Augmented Line Segment Based algorithm with Edge Reversal

3.EXPERIMENTAL RESULTS

The algorithm has been implemented in "C "on GNU/Linux platform. The algorithm was tested on randomly generated set of points. Figure 9 shows the RSMT generated by the Improved Augmented Line Segment Based algorithm with Edge Reversal for a set of 150 random points.

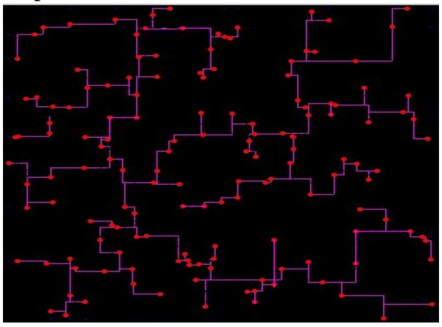

Figure 9. RSMT constructed for a set of 150 random points

Table 1: Comparison of the final length of the tree constructed for RMST, RMST using ALSB and RSMT using Improved ALSB with Edge Reversal

Number of Points	Final length of Rectilinear Minimum Spanning Tree	RSMT total length using ALSB algorithm	RSMT total length using Improved ALSB algorithm with Edge Reversal
50	2497	2321	2303
100	3257	3132	3055
200	4682	4345	4283
300	5849	5173	5145
400	6679	6029	5941
500	7407	6559	6494
600	8054	7105	7012

Table 1 demonstrates the comparison of the total length of the RMST, RSMT using ALSB algorithm [11] and RSMT using Improved ALSB algorithm with Edge Reversal. The algorithm has been tried out with various randomly generated set of points. Table 1 shows that the Improved

ALSB algorithm with Edge Reversal provides a considerable improvement over ALSB algorithm with respect to total RSMT length reduction.

4. CONCLUSION

An improved ALSB with Edge Reversal algorithm has been proposed which provides worthy improvement to the ALSB algorithm in terms of RSMT cost reduction (reduced length). Length reduction is obtained by the reversal of previously selected L-shaped layout whenever the overlap count of rejected edge increases because of the addition of new edge to the RSMT. RSMT length reduction has a significant role in chip design. Efforts are on to apply this algorithm to an industrial benchmark and implement the same with reconfigurable logic.

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., and 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] Ho, J-M., Gopalakrishnan Vijayan, and C. K. Wong. "New algorithms for the rectilinear Steiner tree problem." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 9.2 (1990): 185-193.

[5] Kahng, Andrew B., and Gabriel Robins. "On the performance bounds for a class of rectilinear Steiner tree heuristics in arbitrary dimension." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 11.11 (1992): 1462-1465.

[6] Borah, Manjit, Robert Michael Owens, and Mary Jane Irwin. "An edge-based heuristic for Steiner routing." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 13.12 (1994): 1363-1368.

[7] Chu, Chris, and Yiu-Chung Wong. "FLUTE: Fast lookup table based rectilinear steiner minimal tree algorithm for VLSI design." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 27.1 (2008): 70-83.

[8] Chen, Hongyu, et al. "Refined single trunk tree: a rectilinear steiner tree generator for interconnect prediction." Proceedings of the 2002 international workshop on System-level interconnect prediction. ACM, 2002.

[9] Vani, V., and 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.

[10] Vani, V., and G. R. Prasad. "Augmented line segment based algorithm for constructing rectilinear steiner minimum tree." Communication and Electronics Systems (ICCES), International Conference on. IEEE, 2016.

[11] Vani, V., and G. R. Prasad. "An Improved Augmented Line Segment based Algorithm for the Generation of Rectilinear Steiner Minimum Tree." International Journal of Electrical and Computer Engineering (IJECE) 7.3 (2017).

[12] 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.

AUTHOR

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.