ISSN:0975-9646

A Study on Approximation Algorithms for Constructing Rectilinear Steiner Trees

Latha N.R.

Computer Science and Engineering, Visveswaraiah Technological University
B.M.S. College of Engineering, Bangalore, India.

Abstract— **The Steiner tree problem is one of the complex, combinatorial optimization problems in the graph theory literature. Solving Steiner tree problem is of great importance since its application includes VLSI physical design, routing, wire length estimation etc. The variant of the Steiner tree called Rectilinear Steiner tree is used in the various phases of VLSI design and finding the Minimum Rectilinear Steiner tree is the problem that is in much research. Since the problem is NP-hard, a lot of research is focused in designing good heuristics and approximation algorithms. This paper is a brief study on the different approximation and heuristic algorithms for solving Rectilinear Steiner tree problem.**

Keywords— **Rectilinear Steiner Trees, heuristic algorithms, VLSI physical design.**

I. Introduction

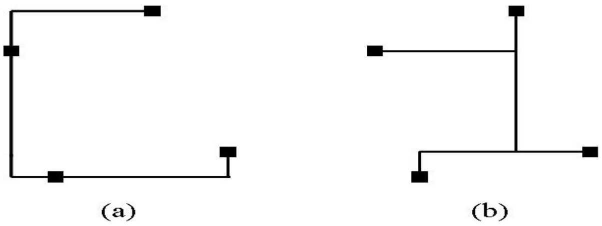
One of the applications of Steiner tree in electronic design automation is in placement and routing phase of VLSI design process. The routing is done in two phases called global routing and detailed routing. In Global routing connections are made between blocks but does not take into account the details of each wire and pin, whereas in detailed routing point-to-point connections between pins on each block is completed.

Given a set of input points, the Steiner tree problem (STP) is to find a minimum-length tree that connects the input points, where new points are added to minimize the length of the tree. These new points are called Steiner points.

One of the variations of Steiner tree problem is the Rectilinear Steiner tree that considers rectilinear or Manhattan distance between a pair of points. The rectilinear distance d (p1, p2) between two points p1 and p2 is defined as: d (p1, p2_=|x1 – x2 | + |y1 – y2 |, where (x_i ,y_i) are the coordinates of pi.

A Rectilinear Steiner Tree (RST) for a set of points P in the plane is a tree which interconnects P using only vertical and horizontal lines. A reduction from RST problem to graph Steiner tree problem allows the use of exact algorithms for graph Steiner tree problem to solve RST problem. Therefore various graph based i.e. spanning tree based algorithms for computing optimal RST are devised. The RST problem is now stated as: Given a set T of n points called Terminals in the plane, find a set of additional points called Steiner points such that the length of a Rectilinear Minimum Spanning Tree (RMST) of the T U S is minimized.

The Minimum Rectilinear Steiner Tree (MRST) problem in the plane is stated as: Given a set P of n points, find a set S of Steiner points such that the minimum spanning tree over P U S has minimum cost. The cost of any edge in the tree is the rectilinear distance between its endpoints, and the cost of a tree is the sum of its edge costs. This is a fundamental problem in global routing and wire estimation for VLSI circuit layout, where we are interested in Steiner trees connecting the pins of a signal net. The RMST and MRST are as shown in Fig. 1



RMST MRST

Fig. 1 Example of an RMST and MRST

Hanan[1] is the first author who considered the rectilinear version of the Steiner tree problem. In 1966 hanan showed that if one draws horizontal and vertical grid lines through each of the points in P, there is an MRST whose Steiner points S are all chosen from among the intersection points i.e., the Steiner candidate set in the resulting grid.

Garey and Johnson[2] worked on the concepts of hanan and showed that even though there is constraint on the solution space, the MRST problem is NP-complete and much effort has been devoted in designing heuristics and approximation algorithms. Garey et al. [2] also proved that no algorithm can compute an Optimal RST in Polynomial time.

This paper is a survey on various heuristic and approximation algorithms for finding Optimal RST's.

II. Spanning tree Based approaches:

M.Hanan [1] is the first to have reduced the RST problem to the Graph Steiner Problem (GST) problem. Hanan's theorem proves that for any instance, an optimal RST exists in which every Steiner point lies at the intersection of two orthogonal lines that contain terminals. Hanan's theorem states that by drawing a graph called Hanan Grid graph, an optimal RST may surely be obtained. The Hanan Grid Graph (G) can be constructed by drawing a Horizontal and Vertical line through each terminal, the vertices in G correspond to the intersection of the lines. There is an edge between the two vertices if they are adjacent along a line and the weight of an edge is the rectilinear distance between its end points. Hanan Grid Graph is as shown in Fig. 2

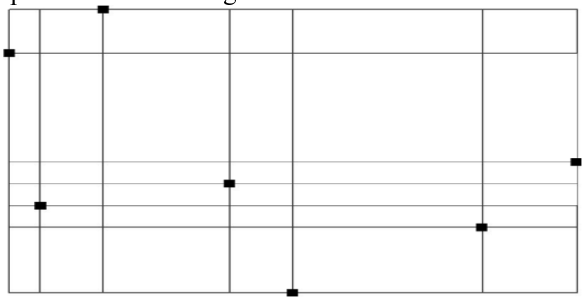


Fig. 1 Hanan Grid Graph

RST for a set A of points is defined as a tree composed of Horizontal and vertical lines which interconnects all members of A. An Optimal RST for A is defined as one in which the lines have the shortest possible total length. The paper specifies that there may be many optimal RST's for A and various algorithms have been proposed for constructing such tree.

Hakimi[3] presented an algorithm for constructing Graph Steiner tree problem called the spanning tree enumeration algorithm. This algorithm considers T points and for every subset of S of n-2 or less number of Non-terminals and computes an MST of T U S and takes the minimum to be the optimal Steiner Tree. The time Complexity of the algorithm is O(n^2 2m).

F.K.Hwang[4] proves that the ratio of the cost of the rectilinear minimum spanning tree (RMST) to that of an optimum RST is no greater than 3/2. Therefore the rectilinear MST is a suitable starting point for deriving low cost RST.

Minimum spanning tree (MST) is defined as a spanning tree of minimal length and a Steiner minimal tree (SMT) is a Steiner tree of Minimal length. l_m is used to denote the length of MST and l_s is used to denote length of SMT and thus proved $l_s \leq l_m$. In this paper it is proved that $l_s/l_m \leq 2/3$ for rectilinear distance.

Ho,Vijayan and CK Wong [5] presented a new approach to construct an RST of a given set of points in the plane starting from a Minimum Spanning tree. The total wire length of a RST is referred as the cost of RST. A heuristic approach is proposed that finds layouts for the edges of the MST so as to maximize the overlaps between the layouts and thus minimizing the cost (i.e. wire length) of the Rectilinear Steiner Tree.

This paper proposes two algorithms for constructing RST from Minimum Spanning Trees, these are optimal under the conditions that the layout of each edge of the MST is (i) An L-Shaped or (ii) Any Staircase Layout respectively. The first algorithm has linear time complexity and the second algorithm has a higher polynomial time complexity. The time complexity is found to O(n) in linear time.

An approach where the edges of Separable MST are replaced by L or Z shaped layout which have maximum overlap is used in this paper. A rectilinear MST of a point set is said to be separable if arbitrary staircase layouts of any two non-adjacent edges of the tree do not intersect or overlap.

Hasan,Wong,vijay[6] propose an heuristic algorithm for constructing a rectilinear Steiner tree (RST) from a rectilinear minimum-spanning tree of a given point set. The heuristic algorithm is based on replacing neighbourhood structures of an independent set of rectilinear minimum-spanning tree points by their optimal RSTs. The time complexity of each iteration of the algorithm is O(n log n), where n is the cardinality of the input point set

Chao Ting hao and Hsu yu chin[7] presented a novel algorithm for constructing a rectilinear Steiner tree (RST) of a given set of points. The algorithm works by incrementally introducing Steiner points from a rectilinear minimum spanning tree (RMST) and generating a refined Steiner tree. Steiner points are introduced in two stages according to their importance and cost. In the first stage, they are generated from the local structure of the tree, in the second stage, they are generated from the global structure using a loop detection method. The proposed algorithm outperforms most other algorithms, the average cost improvement over the rectilinear minimum spanning tree is 10.6% and its time complexity is O(n/sup 2/ log n).

Andrew B. Kahng and Gabriel Robins [8] gives a more direct approach which makes a significant change from spanning tree based strategies by iteratively finding optimal Steiner points to be added to the layout. The method used not only gave improved average-case performance, but also moves away from the examples of previous approaches.

This paper introduces a fast new approach for RST called Iterated-1 Steiner heuristic which repeatedly finds the best possible Steiner point and adds it to the point set until no further improvement is possible, several variations of Iterated – 1 Steiner method are shown, one among them is the Batched Iterated – 1 Steiner heuristic (BI1S).

The approach proposed has a number of advantages (i) The performance ratio of the method is never as bad a 3/2, it is proved that it is not greater than 4/3 on the entire class of instances for which C(MST)/C(MRST) = 3/2, while other known methods have performance ratio arbitrarily close to 3/2. An MST and MRST are as shown in Fig. 3

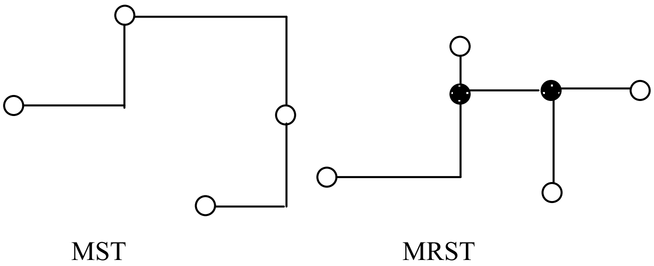


Fig. 3 Example of an MST and MRST

(ii) Average performance is better than previous MST based heuristics, yielding an average improvement of 10% to 11% over MST costs. (iii) The algorithm can be limited so that it introduces only k-Steiner points. (iv) It has many variations or extensions like randomized, batched and parallel variations as well as applications to alternate routing geometrics.
The time complexity of this method is O(n^3) and performance wise this method gives almost 11% average improvement over MST for random point sets and the method is simple. The method also yields results that reduce wire length by upto 2% to 3% over the best previous methods.
This paper directly applies the method of Georgakopoulos and papadimintriou[23] given an O[n^2] method for computing the 1-steiner tree (not just a 1-steiner point) for n points in the Euclidian plane to the Manhattan plane.

M.Borah,R.M. Owens and M.J. Irwin[9] proposed an algorithm for rectilinear Steiner minimal tree that starts from an minimal spanning tree and then iteratively considers connecting a point to a nearby edge and deleting the longest edge. Prims algorithm was used to generate minimal spanning tree in O(n^2) time and sweep line algorithm and least common ancestor algorithm was used to reduce the number of points for finding the longest edge.

Mandoiu and Vazirani[10] proposed that in the extensive survey of RST heuristics upto 1992 by Hwang and Richards the BI1S heuristic of Kahng and Robins emerges as the clear winner with an average improvement over the MST on terminals of almost 11%. Subsequently two other heuristics have been reported to match the same performance.

The main contribution of this paper is a new RST heuristic which is built on the 3/2 approximation algorithm of Rajagopalan et al. [24] for the metric Steiner tree problem on quasi bipartite graphs. These graphs do not contain edges connecting pair of Steiner vertices. The paper gives an RV-based heuristic for finding Steiner trees in arbitrary (non-quasi-bipartite) metric graphs. The heuristic called Iterated RV, computes a Steiner tree of a quasi-bipartite subgraph of the original graph using the RV algorithm, in order to select a set of candidate Steiner vertices.

The RV algorithm is built around the linear programming relaxation of a sophisticated integer program formulation called the bi-directed cut relaxation. This heuristic achieves a good running time by combining an efficient implementation of RV algorithm. This algorithm also runs significantly faster than Robin's implementation of BI1S and geosteiner code for both random and real very large scale integrated instances and also gives high quality solutions.

H.Zhou, N.Shenoy and W.Nicholls[11] introduced the spanning graph as an intermediate step in minimal spanning tree construction. The number of edges in the graph is called the cardinality of the graph and an efficient O(n log n) algorithm to construct a spanning graph of cardinality O(n) is proposed. A sweepline algorithm that efficiently constructs the spanning graph with worst case running time of O(n log n) is proposed by Zhou.

Zhou, Hai[12] proposed an implementation of an efficient Steiner minimal tree algorithm that has a worst case running time of O(n log n) and a similar performance of iterated -1 Steiner algorithm. The algorithm efficiently combines Borah et al [21] edge substitution concept with Zhou et al [22] spanning graph. The algorithm first generates an initial minimal spanning tree and then to generate point-edge pairs for tree improvements using spanning graph.

Kahng, Mandoiu and Zelikovsky [13] proposed a O($n\log^2 n$) batched version of greedy triple contraction algorithm [20], Batched Greedy Algorithm [BGA].
In BGA a batch of triples (triple is the optimal full Steiner tree for a set of three points where all the points are leaves) are added in each iteration to construct RSMT. The Gain of adding a triple is computed by adding two edges for connecting the three points of a triple (addition of triple) and removing the longest edges in the formed loop (contraction). All the triples with positive gain are identified and added in a non-decreasing order of gain by addition of a triple and then contraction. The Steiner points which are formed by the triple are added to the set of points P and MST over P computed iteratively.

Cinel and Bazlamacci [14] proposed a distributed RSMT algorithm based on the idea of Kahng et al. [13] Batched Greedy Algorithm (BGA) and Zhou et al. [19] Rectilinear Spanning graph (RSG) algorithm. The modified algorithm used RSG approach to generate initial sparse graph and to update the MST. BGA approach was used for longest edge computation. Distributed version was based on parallelization of major components of the modified RSMT.

III. LOOK-UP TABLE BASED APPROACHES:

Chris Chu[15] presented a novel wirelength estimation called FLUTE. This technique is based on pre-computed lookup table to make wirelength estimation very fast and very accurate for low degree nets. Two techniques for wirelength estimation are compared one using Half Perimeter Wirelength (HPWL) and the other based on rectilinear spanning tree.
In this look-up table based wirelength estimation the set of all degree-n nets are partitioned into n! groups according to the relative positions of their pins. For each group, the wirelength of all possible routing topologies are written as a small number of linear combinations of distances between adjacent pins. This linear combination is called as a potentially optimal wirelength vector (POWV). Redundant POWV's are removed and rest are stored for each group into the table. To find optimal wirelength of a net the wirelengths corresponding to POWV's for the group the

net belongs to is to be computed and the one with minimum wirelength is selected.
This technique works well with low-degree nets, for high-degree nets a net-breaking technique is proposed to reduce the net size until the table can be used. The net-breaking direction can be either horizontal or vertical. The two subnets are then routed independently.

Chu and Wong [16] proposed a fast look-up table based O(nlogn) algorithm(FLUTE) for RSMT construction which is an extension of the wirelength estimation by Chris Chu[15]. The main contribution of this paper is a new net breaking technique which is much better than the one in Chris Chu[15]. Three different net breaking heuristics are proposed. Experimental results showed that the quality of FLUTE improved with accuracy parameter value and produced better solution than BI1S but with increased runtime. The average wirelength error over all nets is only 0.07%

Chris Vhu,Yiu Chung Wong[17] proposed a technique which showed improvement of FLUTE over the other previous works. The previous techniques were very fast and accurate RSMT algorithms but only upto degree 9 (also upto degree 30) and was not effective for higher degrees. The main idea is to partition a net according to a spanning tree into small subnets which the original flute can handle and then to merge the Steiner trees for the subnet together so that the error introduced is minimized.
The resulting algorithm FLUTE 3.0 has a runtime complexity of O(nlog^{2}n) and is extremely fast and accurate in practice. The accuracy time trade-off achieved is better for degree 10 or more and also 30 or more. FLUTE 3.0 is also highly scalable, for instance it can route a 1,00,000 pin net in less than 1 minute and 1 million-pin net in less than 10 minutes and a 3-million pin net in about 25 minutes

Chris Vhu,Yiu Chung Wong[18] present a very fast and accurate RSMT algorithm called FLUTE. The technique followed is the same as previous versions of flute that are based on pre-computed lookup table to make RSMT construction very fast and very accurate for low-degree nets. The set of all degree-n nets are partitioned into n! groups according to the relative positions of their pins.
For each group, vector called Potentially Optimal Wirelength Vector (POWV) are found. The POWV's for each group is precomputed and stored in a table. One corresponding Steiner tree, called as Potentially Optimal Steiner Tree (POST) associated with the POWV with minimum wirelength is also stored. For high –degree nets, a net breaking technique is proposed to reduce the net size until the table can be used. This paper proposes a scheme that allows users to control the trade-off between accuracy and runtime. The runtime complexity is O(n^2) and the average wire length error over all nets is only 0.075%. For higher degree nets, FLUTE with a small accuracy parameter (A) value can generate reasonably accurate solutions in a very short time.

Spanning Tree Based approaches			
Sl. No	Heuristic	Time Complexity	Technique and Results
1	Hasan et al[6]	O(n log n)	Replacing neighbourhood structures
2	Andrew et al[8] (BI1S)	O(n^3)	Based on Iterated-1 Steiner and produces near-optimal solution High runtime
3	Zhou et al[12] (RSG)	O(n log n)	Based on edge substitution and has good runtime
4	Khang et al[13] (BGA)	O(nlog^{2}n)	Based on Batched Greedy Triple Contraction, produces near-optimal solution giving a good runtime
Look-up Table Based Approaches			
1	Chu et al[16] FLUTE	O(n log n)	Fast look-up table based, the solution is not so optimal for larger nets.
2	Chu et al[17] FLUTE	O(n log^2 n)	Fast look-up table based , uses net breaking technique for large nets, produces near optimal results with increased runtime
3	Chu et al[18] FLUTE	O(n^2)	Fast look-up table based, uses Accuracy parameter to show tradeoff between runtime and accuracy, gives good runtime and more efficiency.

Table 1: Comparison of Important Approximation algorithms for the construction of RSMT

IV. CONCLUSIONS

This paper is a brief study of the existing heuristic and approximation algorithms for computing the Rectilinear Steiner Minimum Trees. Table 1 gives the comparison of the existing important algorithms for the construction of RSMT. There are various parameters that differentiate one heuristic with the other, important ones being reduced wirelength that yields optimal solution and reduced runtime. Some algorithms that gives reduced wirelength are Kahng et al[13] and Zhou et. al[12] and if runtime is the main concern, then Chu et al[18] is the best option. It is observed that some algorithms while giving better solution quality takes more runtime and some have reduced runtime but increased wirelength, thus the choice of algorithm is a compromise between wirelength and runtime.

ACKNOWLEDGEMENT

The author would like to acknowledge and thank Technical Education Quality Improvement Program [TEQIP] Phase 2, BMS College of Engineering.

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] Steiner's problem in graphs and its implications S. L. Hakimi

[4] Hwang, Frank K. "On Steiner minimal trees with rectilinear distance." SIAM journal on Applied Mathematics 30.1 (1976): 104-114.

[5] 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.

[6] Hasan, N., G. Vijayan, and C. K. Wong. "A neighborhood improvement algorithm for rectilinear Steiner trees." Circuits and Systems, 1990. IEEE International Symposium on. IEEE, 1990.

[7] Ting-Hai, Chao, and Hsu Yu Chin. "Rectilinear Steiner tree construction by local and global refinement." Computer-Aided Design, 1990. ICCAD-90. Digest of Technical Papers. 1990 IEEE International Conference on. IEEE, 1990.

[8] Kahng, Andrew B., and 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.

[9] M.Borah,R.M. Owens and M.J. Irwin."An edge based heuristic for steiner routing "Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on VOL 13,No.12,1994

[10] Mandoiu, Ion I., Vijay V. Vazirani, and Joseph L. Ganley. "A new heuristic for rectilinear Steiner trees." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 19.10 (2000): 1129-1139.

[11] H. Zhou, N. Shenoy, and W. Nicholls, "Efficient spanning tree construction without Delaney triangulation", Information Processing Letter, 2002, 81(5): pp.-.

[12] Zhou, Hai. "Efficient Steiner tree construction based on spanning graphs." Proceedings of the 2003 international symposium on Physical design. ACM, 2003.

[13] Kahng, Andrew B., Ion I. Mandoiu, and 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.

[14] Cinel, Sertaç, and C. F. Bazlamacci. "A Distributed Heuristic Algorithm for the Rectilinear Steiner Minimal Tree Problem." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 27.11 (2008): 2083-2087.

[15] Chris Chu. FLUTE: Fast lookup table based wirelength estimation Technique. In Proc. IEEE/ACM Intl. Conf. on Computer-Aided Design,
pages 696–701, 2004.

[16] Chris Chu and Yiu-Chung Wong. Fast and accurate rectilinear Steiner
minimal tree algorithm for VLSI design. In Proc. Intl. Symp. on Physical Design, pages 28–35, 2005

[17] Wong, Yiu-Chung, and 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.

[18] 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.

[19] Zhou, Hai. "Efficient Steiner tree construction based on spanning graphs." Proceedings of the 2003 international symposium on Physical design. ACM, 2003.

[20] Zelikovsky, Alexander Z. "An 11/6-approximation algorithm for the network Steiner problem." Algorithmica 9.5 (1993): 463-470.

[21] 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): 1563-1568.

[22] Zhou, Hai, Narendra Shenoy, and William Nicholls. "Efficient minimum spanning tree construction without Delaunay triangulation." Proceedings of the 2001 Asia and South Pacific Design Automation Conference. ACM, 2001.

[23] G. Georgakopoulos and C. Papadimitriou. The 1-Steiner tree problem. J. Algorithms 8 (1987) 122–130.

[24] Rajagopalan, Sridhar, and Vijay V. Vazirani. "On the bidirected cut relaxation for the metric Steiner tree problem." Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 1999