A Polynomial Time Approximation Scheme for Rectilinear Steiner Minimum Tree Construction in the Presence of Obstacles

Jian Liu Ying Zhao Eugene Shragowitz George Karypis

Department of Computer Science and Engineering
University of Minnesota
USA

ABSTRACT

One problem in VLSI physical designs is to route multi-terminal nets in the presence of obstacles. This paper presents a polynomial time approximation scheme for construction of a rectilinear Steiner minimum tree in the presence of obstacles. Given any m rectangular obstacles, n nodes and $\varepsilon > 0$, the scheme finds a $(1+\varepsilon)$-approximation to the optimum solution in the time $n^{o(1/\varepsilon)}$, providing an alternative of previous heuristics

Remark: m is assumed to be a constant, otherwise when we solve the sub-problem in a brute force manner, we cannot declare that it can be solved in constant time.

Keywords Rectilinear Steiner Minimum Tree in presence of obstacles, VLSI routing, PTAS, Guillotine cut, approximation algorithm.

1. INTRODUCTION

One important issue in VLSI physical design is routing a net that connects multiple terminals. The routing process practically is conducted in the presence of obstacles. These obstacles are occupied by either logic blocks or wires in the previous routed nets. This problem began to draw more and more attention [1][2]. Even in the absence of obstacles, finding a rectilinear Steiner minimum tree is NP complete [3]. It implies that finding a RSMT in the presence of obstacles cannot be solved in polynomial time exactly since the introduction of obstacles even makes it more difficult to the find the minimum distance between two points.

Our paper presents a polynomial time approximation scheme for construction of a rectilinear Steiner minimum tree in the presence of obstacles. The rest of the paper is organized as following. In section 2, the problem formulation is given. Section 3 provides a preliminary of Guillotine cut techniques. The algorithms are described in the Section 4 and the time complexity is analyzed. The conclusions and future work is addressed in the section 5.

2. PROBLEM FORMULATION

This problem arises from VLSI routing phase. A net connecting multiple terminals is expected to be connected in a shortest tree which avoids obstacles like wires and/or vias (see Fig. 1). Without any loss of generality, we assume that all obstacles are rectangles as a rectangular obstacle can be decomposed into a set of rectangles. In VLSI layout, the space requirements between the boundaries of obstacles and nets are applied. Each rectangular is expanded in four edges with the length d. After this expansion, the nets are allowed to overlap with the boundary of obstacles.

The problem is then formulated as following: Given *m* rectangular obstacles and *n* terminals on one plane, the goal is to find a shortest rectilinear tree that interconnects the terminals without intersecting the interior of any obstacle.

In the following sections, we demonstrate such problems can be solved using the Guillotine-cut technique [5][6].

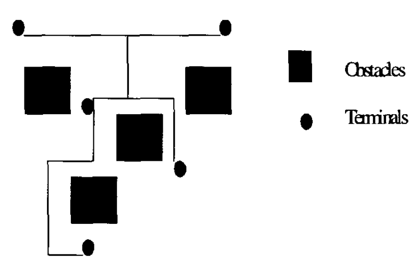

Figure 1: A rectilinear Steiner Tree in presence of Obstacles

3. GUILLOTINE CUT PRELIMINARIES

Guillotine-cut technique is a strong approach that establishes a framework to solve a class of geometric optimization problems. It has been applied to solve different problems [8][9]. In the section, the fundamental ideas of this methodology are summarized as follows. The guillotine subdivision technique is introduced initially to solve rectangular partition problems in [5] [6]. For simplicity and convenience, notations similar to those in [10] are used.

Definition m-dark point: Given a partition, a point p is a horizontal (vertical) m-dark point if the horizontal (vertical) line passing through p intersects at least m vertical segments on the left of p (above p) and at least m vertical segments on the right of p (below p).

Definition m-guillotine cut: A horizontal (vertical) cut is an m-guillotine cut if it consists of horizontal (vertical) m-dark points on the cut line.

Let $H_m(V_m)$ denotes the sets of all horizontal (vertical) m-dark points, the following lemma can be proved.

Lemma 1: There exists either a horizontal line L such that $length(L \cap H_m) \le length(L \cap V_m)$ or a vertical line L such that $length(L \cap H_m) \ge length(L \cap V_m)$

This lemma implies that there always exists an m-guillotine cut. As the cost of k-guillotine cut can be symmetric charged to the k segments left/right sides or upper/lower sides. The increased cost by added k-Guillotine cuts is bounded by 1/k of total cost of charged segment. It also can be proved that there exist such horizontal (vertical) cut lines passing through the vertices of the partition, or the midpoints between the vertices. Based on this lemma, the following theorem can be further proved.

Theorem 1: Every rectangular partition P can be modified into an m-guillotine rectangular partition P' with total length

$$length(P') \le (1 + 1/m) length(P)$$

The above theorem means that any partition can be transformed into a sequence of *k*-guillotine cut partition until it can be the rectangle contains <2*k* terminals. In this case, the one optimal solution can be solved in brute force manner. M guillotine cut features dynamic programming. If all the possible divisions are explored exhaustively, the optimal solution is bounded by (1+1/*k*) OPT. To assure (1+1/*k*) <= $(1+\varepsilon)$ which is the expected performance ratio, *k*> $(1/\varepsilon)$ is required.

Based on the *m*-guillotine techniques, PTASs for various geometric optimization problems can be solved, such as TSP, SMT, and K-MST etc. The definition of m-guillotine cut may vary to accommodate the subject. In the following text, we demonstrate how this technique can be applied to solve the above routing problem.

4. ALGORITHM AND COMPLEXITY ANALYSIS

4.1. The existence of PTAS

It is proved in[1] [3][4] that obstacle-avoiding Steiner trees are solvable on the extended Hanan grids induced by the input terminals and obstacles. The extended Hanan grids (Fig. 2) are the vertical and horizontal lines passing through each terminal and the escape lines bounding each obstacle. Give n terminals and m obstacles, the size of the extended Hanan grids can be bounded by O(n+m). As proved in [1][4], there exists an optimal RSMT only containing segments on the extended Hanan grids. The proof can be summarized as a sequence of shifts. If there are segments that do not lie on the extended Hanan grids, then these segments can be shifted left or right (up or down) until they hit the extended Hanan grids and the resulting RSMT is still an optimal RSMT.

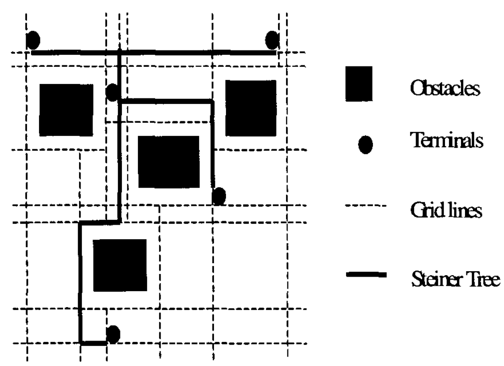

Figure 2: A rectilinear Steiner Tree in presence of Obstacles on the Hanan grids

Theorem 2:
The obstacle-avoiding Rectilinear Steiner Tree problem has a $(1+1/k)$ PTAS (Polynomial Time Approximation Scheme).

Proof: Let P be a set of n terminals and let O be a set of m obstacles in the plane. Let T be an optimal OARSMT for P and O with length *L*. Without lose of generality, we assume that T lies on the extended Hanan Grid. We then show that T can be modified to be a *k*-guillotine OARSMT T' with the total length less than (1+1/*k*)*L*. An OARSMT can be treated as a rectilinear partition with all the edges as cut lines. A *k*-guillotine OARSMT contains a sequence of *k*-guillotine cuts that partition the whole area adaptively until the resulting rectangles only contain one terminal. The modification of T to T' are the following. Each time, we consider the cut lines passing through the midpoint of the extended Hanan Grids. By lemma 1, there exists a *k*-guillotine cut whose length can be symmetrically charged off to parallel 2*k* edges in T, in which each edge is charged off 1/2*k* of the length. Then we add all the segments in this *k*-guillotine cut into T. Since every edge in T will not be charged more than once from one direction, the total added length of *k*-guillotine cuts is at most $1/k$ L. We call the resulting rectilinear partition E, which contains all the edges in T and all newly added k-guillotine cuts. Adding new edges to T may introduce loops. By removing edges to break loops and adjusting edges to remove non-terminal endpoints, E can be modified to a feasible OARSMT T' with length less then the length of E. Thus, T' is a k-guillotine OARSMT with length less than $(1+1/k)L$.

4.2. Dynamic programming

An optimal k-guillotine OARSMT (obstacle-avoiding Rectilinear Steiner Minimum Tree) for n terminals and m obstacles can be found by applying dynamic programming. The running time of the dynamic programming is $O((n+m)^{10k+5}2^{O(k+m)})$ and the optimal k-guillotine OARSMT is (1 + 1/k)-approximation for the OARSMT.

Given an OARSMT problem with n terminals and m obstacles, we now consider the extended Hanan grids plus all the midpoint grids. The x and y coordinates of these grids determine all-possible sub rectangles to find the optimal solution.

The subproblem of this dynamic programming algorithm is defined as follows: given a rectangle with boundary conditions and a connection pattern, compute a minimum length m-guillotine obstacle-avoiding rectilinear Steiner forest, such that the resulting edge set E of the forest is an m-guillotine partition and connects terminals inside the rectangle with crosspoints and segments on boundaries, according to the connection pattern, without passing through obstacles.

The number of sub rectangles is bounded by $O(n+m)^4$, since the corners of the boundary box are chosen from the intersections of the extended Hanan grids plus all the midpoint grids. The boundary conditions are given in the following way. For each boundary, there are at most $2k$ crosspoints and at most m k-guillotine cut segments, noticing that k-guillotine subdivision does not increase the number of obstacles in the either of the sub-problem. One k-guillotine cut may consist of segments, since there are obstacles present. Once the crosspoints for the k-guillotine cut are finalized, the k-guillotine cut segments are determined by input obstacles. Thus, we only exhaustively enumerate the possible position of the crosspoints. The number of possible boundary conditions can be bounded by $O((n+m)^{8k})$. The connection pattern is a partition of the crosspoints and k-guillotine cut segments, such that the crosspoints and k-guillotine cut segments can be connected within one set without crossing the connecting edges of other sets. The number of possible such partitions is $O(2^{O(k+m)})$. Thus the total number of the subproblems is $O((n+m)^{8k+4}2^{O(k+m)})$.

Remark: when we partition the rectangle into the two subdivisions, each obstacle either falls into one of them as whole, or be partitioned into 2 obstacles, each of them falls into one subdivisions, the number of obstacles will not increase for the sub problems.

The base sub-problem, i.e., rectangles that only contain less than one terminal, can be solved in a brute force manner. Other sub-problems can be solved recursively by splitting the problem into two child problems using k-guillotine cuts, and optimizing over all choices of m-guillotine splits. When splitting the problem into two child problems, connection patterns for the two child problems are determined with no conflict to the connection pattern of the original sub-problem. The all-possible child problems splitting can be seen in the following steps: choosing a cut line position within the rectangle, locating at most $2k$ crosspoints at the cut line, and determining connection patterns for the child problem of the crosspoints and at most m k-guillotine cut segments on the selected m-guillotine cut. Thus, each sub-problem can be solved in $O((n+m)^{2k+1}2^{O(k+m)})$.

Combining the number of all-possible sub-problems $O((n+m)^{8k+4}2^{O(k+m)})$ and the running time for one sub-problem $O((n+m)^{2k+1}2^{O(k+m)})$, we have the total running time of this dynamic programming algorithm $O((n+m)^{10k+5}2^{O(k+m)})$.

5. CONCLUSIONS AND FUTURE WORK

Guillotine cuts and portals are two important techniques in designing PTAS. In this work, we adopt Guillotine cut technique straightly to solve this problem. Our approximation scheme only gives a polynomial time algorithm with respect to the number of terminals. When we consider the number of obstacles also as the input size, the running time of the proposed dynamic programming is not polynomial any more. Our next step research will focus on eliminating this limitation. Also we will study how to combine the portals techniques [11] with guillotine cut to improve the computation complexity.

ACKNOWLEDGEMENT

We gratefully thank 1) Dr. D. Z Du for the valuable discussions and suggestion, 2) Dr. B. Lu for providing useful pointers.

12. REFERENCES

[1] J. L. Ganley and J. P. Cohoon, "Routing a Multi-Terminal Critical Net: Steiner Tree Construction in the Presence of Obstacle", *IEEE International Symposium on Circuits and Systems, pp. 113-116*, 1994.

[2] E. Shragowitz and J. Liu "Generation of Minimal Delay Routing Trees in Presence of Obstructions" *pp. 251-254, Proceedings of European Conference on Circuit Theory*

and Design (ECCTD '99) 29 August - 2 September, 1999 Stresa, Italy.

[3] M. R. Garey and D. S. Johnson. "The rectilinear Steiner tree problem is NP-Complete", *SIAM Journal on Computing, 16:1004-1022,* 1987.

[4] M. Zachaiasen, "A Catalog of Hanan Grid Problems", *Networks*, volume 38, Issue 2, pages 76-83, 2001.

[5] J.S.B. Mitchell, "Guillotine subdivisions approximate polygonal subdivisions: A simple new method for geometric k-MST problem", *Proc. 7th ACM-SIAM Symposium on Discrete Algorithms,* (1996) pp 402-408.

[6] J.S.B. Mitchell, "Guillotine subdivisions approximate polygonal subdivisions: Part II – A simple polynoimal-time approximation scheme for geometric k-MST, TSP, and related problems", *SIAM J. Computer.* 29 (1999) no. 2, 515-544.

[7] Naveed Sherwani, "*Algorithms for VLSI Physical Design Automation*", 3rd edition, Kluwer Academic Publishers, 1999.

[8] X. Cheng, B. DasGupta, and B. Lu, "A Polynomial Time Approximation Scheme for the Symmetric Rectilinear Steiner Arborescence Problem", accepted by *Journal of Global Optimization.*

[9] B. Lu and L. Ruan, "Polynomial Time Approximation Schemes for the Rectilinear Steiner Arborescence Problem", *Journal of Combinational Optimization*, Vol 4, No. 3, September 2000, pp 357-363.

[10] Ding-Zhu Du and Ker-I Ko, "*Design and Analysis of Approximation Algorithms*", (in preparation).

[11] S. Arora, "Polynomial time approximation schemes for Euclidean TSP and other geometric problems," *Journal of ACM*