Guillotine Cut in Approximation Algorithms

Xiuzhen Cheng* Ding-Zhu Du* Joon-Mo Kim* Hung Quang Ngo *

April 18, 2001

Abstract

The guillotine cut is one of main techniques to design polynomial-time approximation schemes for geometric optimization problems. This article is a mini survey on its history and current developments.

*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA. E-mail: {cheng,dzd,jkim,hngo}@cs.umn.edu.

1 Introduction

In 1996, Arora [1] published a surprising result that many geometric optimization problems, including the Euclidean TSP (traveling salesman problem), the Euclidean SMT (Steiner minimum tree), the rectilinear SMT, the degree-restricted-SMT, k-TSP, and k-SMT, have polynomial-time approximation schemes. More precisely, for any $\varepsilon > 0$, there exists an approximation algorithm for those problems, running in time $n^{O(1/\varepsilon)}$, which produces approximation solution within $1 + \varepsilon$ from optimal. It made Arora's research be reported in *New York Times* again. [1] Several weeks later, Mitchell [19] claimed that his earlier work [17] (its journal version [18]) already contains an approach which is able to lead to the similar results. However, one year later, Arora [2] made another big progress that he improved running time from $n^{O(1/\varepsilon)}$ to $n^3(\log n)^{O(1/\varepsilon)}$. His new polynomial-time approximation scheme also runs randomly in time $n(\log n)^{O(1/\varepsilon)}$. Soon later, Mitchell [20] claimed again that his approach can do a similar thing. We were curious about this piece of history and hence made a study on these two approaches. In this article, we would like to share with readers the result of our investigation and something interesting that we found in their publications.

2 Rectangular Partition and Guillotine Cut

Let us start from rectangular partition. In fact, before prove his main theorem, Mitchell [17, 18] stated clearly that "Our proof is inspired by the proof in [7]" where the reference [7] in [17] ([9] in [18]) is actually a paper of Du, Pan, and Shing [7] on minimum edge-length rectangular partition. This paper initiated the idea of using guillotine cut to design approximation algorithms.

The minimum edge-length rectangular partition (MELRP) was first proposed by Lingas, Pinter, Rivest, and Shamir [13]. It can be stated as follows: Given a rectilinear polygon possibly with some rectangular holes, partition it into rectangles with minimum total edge-length.

[1] Arora had his first work reported in *New York Times* in 1992 about probablistic checkable proof system.

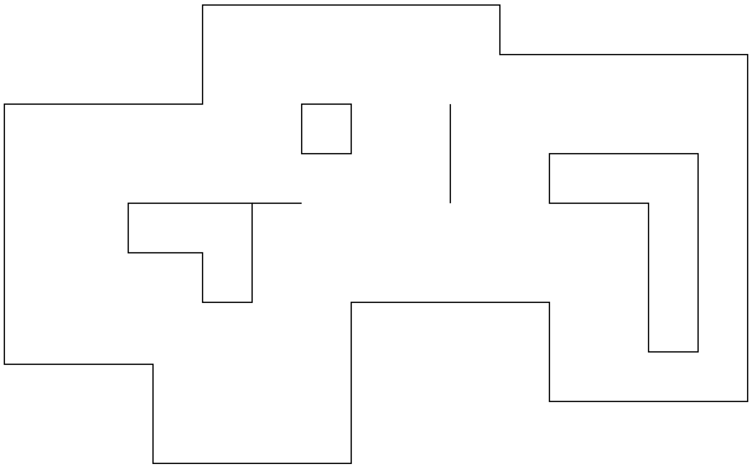

Figure 1: Rectilinear polygon with holes.

The holes in the input rectangular polygon can be, possibly in part, degenerated into a line segment or a point (Fig. 1).

There are several applications mentioned in [13] for the background of the problem: “Process control (stock cutting), automatic layout systems for integrated circuit (channel definition), and architecture (internal partitioning into offices). The *minimum edge-length* partition is a natural goal for these problems since there is a certain amount of waste (e.g. sawdust) or expense incurred (e.g. for dividing walls in the office) which is proportional to the sum of edge lengths drawn. For VLSI design, this criterion is used in the MIT ‘PI’ (Placement and Interconnect) System to divide the routing region up into channels - we find that this produces large ‘natural-looking’ channels with a minimum of channel-to-channel interaction to consider.”

They showed that the holes in the input make difference on the computational complexity. While the MELRP in general is NP-hard, the MELRP for hole-free inputs can be solved in time $O(n^4)$ where n is the number of vertices in the input rectilinear polygon. The polynomial algorithm is essentially a dynamic programming based on the following fact.

Through each vertex of the input rectilinear polygon, draw a vertical line and a horizontal line. Those lines will form a grid in the inside of the rectilinear polygon. Let us call this grid the *basic grid* for the rectilinear polygon (Fig. 2).

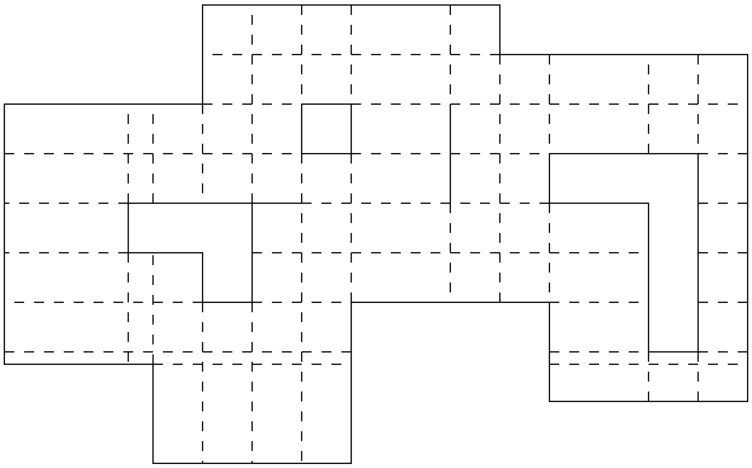

Figure 2: Basic grid.

Lemma 2.1 *There exists an optimal rectangular partition lying in the basic grid.*

Proof. Consider an optimal rectangular partition not lying in the basic grid. Then there is an edge not lying in the basic grid. Consider the maximal straight segment in the partition, containing the edge. Say, it is a vertical segment ab. Suppose there are r horizontal segments touching the interior of ab from right and l horizontal segments touching the interior of ab from left. If $r \geq l$, then we can move ab to the right without increasing the total length of the rectangular partition. Otherwise, we can move ab to the left. We must be able to move ab into the basic grid because, otherwise, ab would be moved to overlapping with another vertical segment, so that the total length of the rectangular partition is reduced, contradicting the optimality of the partition. □

A naive idea to design approximation algorithm for general case is to use a forest connecting all holes to the boundary and then to solve the resulting hole-free case in $O(n^4)$ time. With this idea, Lingas [14] gave the first constant-bounded approximation; its performance ratio is 41. Later, Du [9, 10] improved the algorithm and obtained a approximation with performance ratio 9. Meanwhile, Levcopoulos [15] provided a greedy-type faster approximation with performance ratio 29 and conjectured that his approximation may have performance ratio 4.5.

Motivated from a work of Du, Hwang, Shing, and Witbold [6] on application of dynamic programming to optimal routing trees, Du, Pan, and Shing [7] initiated an idea which is important not only to the MELRP problem, but also to many other geometric optimization problems. This idea is about guillotine cut. A cut is called a *guillotine cut* if it breaks a connected area into at least two parts. A rectangular partition is called a *guillotine rectangular partition* if it can be performed by a sequence of guillotine cuts. Du *et al* [7] noticed that there exists a minimum length guillotine rectangular partition lying in the basic grid, which can be computed by a dynamic programming in $O(n^5)$ time. Therefore, they suggested to use the minimum length guillotine rectangular partition to approximate the MELRP and tried to analyze the performance ratio. Unfortunately, they failed to get a constant ratio in general and only obtained a result in a special case.

In this special case, the input is a rectangle with some points inside. Those points are holes. It had been showed (see [11]) that the MELRP in this case is still NP-hard. Du *et al* [7] showed that the minimum length guillotine rectangular partition as approximation of the MELRP has performance rato at most 2 in this special case. The following is a simple version of their proof, published in [8].

Theorem 2.2 *The minimum length guillotine rectangular partition is a approximation with performance ratio 2 for the MELGP.*

Proof. Consider a rectangular partition P. Let $proj_x(P)$ denote the total length of segments on a horizontal line covered by vertical projection of the partition P.

A rectangular partition is said to be covered by a guillotine partition if each segment in the rectangular partition is covered by a guillotine cut of the latter. Let $guil(P)$ denote the minimum length of guillotine partition covering P and $length(P)$ the total length of rectangular partition P. We will prove

$$guil(P) \leq 2 \cdot length(P) - proj_x(P)$$

by induction on the number k of segments in P.

For $k = 1$, we have $guil(P) = length(P)$. If the segment is horizontal, then we have $proj_x(P) = length(P)$ and hence

$$guil(P) = 2 \cdot length(P) - proj_x(P).$$

If the segment is vertical, then $proj_x(P) = 0$ and hence

$$guil(P) < 2 \cdot length(P) - proj_x(P).$$

Now, we consider $k \geq 2$. Suppose that the initial rectangle has each vertical edge of length a and each horizontal edge of length b. Consider two cases:

Case 1. There exists a vertical segment s having length $\geq 0.5a$. Apply a guillotine cut along this segment s. Then the remainder of P is divided into two parts P_1 and P_2 which form rectangular partition of two resulting small rectangles, respectively. By induction hypothesis,

$$guil(P_i) \leq 2 \cdot length(P_i) - proj_x(P_i)$$

for $i = 1, 2$. Note that

$$\begin{array}{rcl} guil(P) & \leq & guil(P_1) + guil(P_2) + a, \\ length(P) & = & length(P_1) + length(P_2) + length(s), \\ proj_x(P) & = & proj_x(P_1) + proj_x(P_2). \end{array}$$

Therefore,

$$guil(P) \leq 2 \cdot length(P) - proj_x(P).$$

Case 2. No vertical segment in P has length $\geq 0.5a$. Choose a horizontal guillotine cut which partitions the rectangle into two equal parts. Let P_1 and P_2 denote rectangle

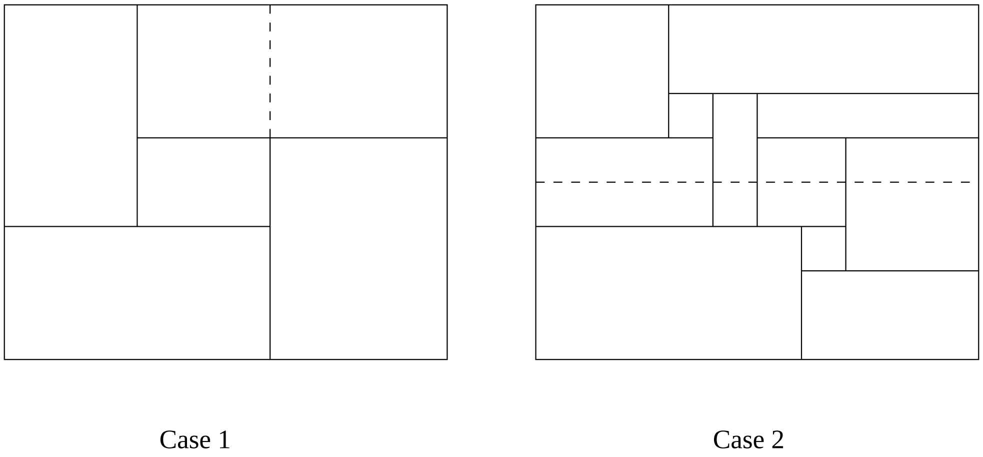

Figure 3: The proof of Theorem 2.2.

partitions of the two parts, obtained from P. By induction hypothesis,

$$guil(P_i) \leq 2 \cdot length(P_i) - proj_x(P_i)$$

for $i = 1, 2$. Note that

$$\begin{aligned} guil(P) &= guil(P_1) + guil(P_2) + b, \\ length(P) &\geq length(P_1) + length(P_2), \\ proj_x(P) &= proj_x(P_1) = proj_x(P_2) = b. \end{aligned}$$

Therefore,

$$guil(P) \leq 2 \cdot length(P) - proj_x(P).$$

$\square$

Gonzalez and Zheng [12] improved the constant 2 in Theorem 2.2 to 1.75 with a very complicated case-by-case analysis. Du, Hsu, and Xu [8] extended the idea of guillotine cuts to the convex partition problem.

3 1-Guillotine Cut

Mitchell [17, 18] gave an approximation with performance ratio 2 for the MELRP in the general case by extending the idea of guillotine cut.

First, he uses a rectangle to cover the input rectangular polygon with holes. Then, he extended the guillotine cut to the 1-guillotine cut. A *1-guillotine cut* is a partition of a rectangle into two rectangles such that the cut line intersects considered rectangular partition with at most one segment (Fig. 4). For simplicity, the length of this segment is called the length of the 1-guillotine cut. A rectangular partition is 1-guillotine if it can be realized by a sequence of 1-guillotine cuts (Fig. 5). The minimum 1-guillotine rectangular partition can also be computed by dynamic programming in $O(n^{16})$ time. In fact, at each step, the 1-guillotine cut has $O(n^4)$ choices. There are $O(n^4)$ possible rectangles appearing in the algorithm. Each rectangle has $O(n^8)$ possible boundary conditions.

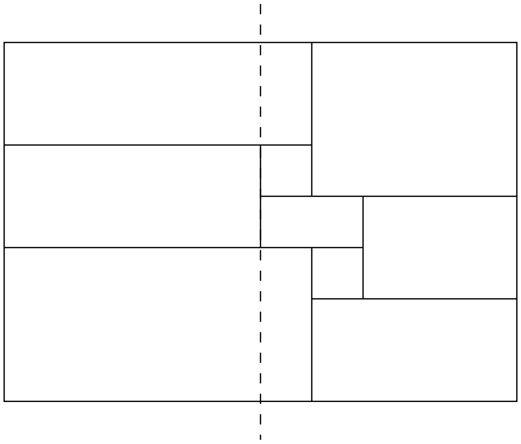

Figure 4: 1-guillotine cut.

To establish the performance ratio of the minimum 1-guillotine rectangular partition as an approximation of the MELRP, Mitchell [17] showed the following.

Theorem 3.1 *For any rectangular partition P, there exists a 1-guillotine rectangular par-*

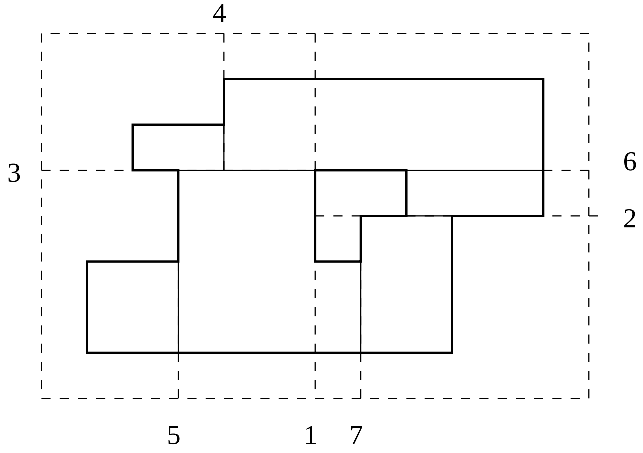

Figure 5: 1-guillotine rectangular partition with seven cuts.

tition P' covering P such that

$$length(P') \leq 2length(P).$$

Proof. It can be proved by an argument similar to the proof of Theorem 2.2. Let $guil_1(P)$ denote the minimum length of a 1-guillotine rectangular partition covering P and $length(P)$ the length of the rectangular partition P. Let $proj_x(P)$ ($proj_y(P)$) denote the total length of segments on a horizontal (vertical) line covered by vertical (horizontal) projection of the partition P We will prove

$$guil_1(P) \leq 2 \cdot length(P) - proj_x(P) - proj_y(P)$$

by induction on the number k of segments in P.

For $k = 1$, we have $guil_1(P) = length(P)$. Without loss of generality, assume that the segment is horizontal. Then we have $proj_x(P) = length(P)$ and $proj_y(P) = 0$. Hence

$$guil_1(P) = 2 \cdot length(P) - proj_x(P) - proj_y(P).$$

Now, we consider $k \geq 2$ in the following two cases:

Case 1. There exists a 1-guillotine cut. Without loss of generality, assume this 1-guillotine cut is vertical with length a. Suppose the remainder of P is divided into two parts P_1 and P_2. By induction hypothesis,

$$guil_1(P_i) \le 2 \cdot length(P_i) - proj_x(P_i) - proj_y(P_i)$$

for $i = 1, 2$. Note that

$$\begin{aligned} guil_1(P) &\le guil_1(P_1) + guil_1(P_2) + a, \\ length(P) &= length(P_1) + length(P_2) + a, \\ proj_x(P) &= proj_x(P_1) + proj_x(P_2) \\ proj_y(P) &\le proj_y(P_1) + proj_y(P_2). \end{aligned}$$

Therefore,

$$guil_1(P) \le 2 \cdot length(P) - proj_x(P) - proj_y(P).$$

Case 2. There does not exist 1-guillotine cut. In this case, we need to add a segment to partition P such that the resulting partition has a 1-guillotine cut and the length of added segment is at most $proj_x(P_1)+proj_x(P_2)-proj_x(P)$ if the 1-guillotine cut is horizontal and at most $proj_y(P_1) + proj_y(P_2) - proj_y(P)$ if the 1-guillotine cut is vertical, where P_1 and P_2 are partitions obtained from P by the 1-guillotine cut. To do so, it suffices to show that there exists a line such that the length of added segment for the line to become a 1-guillotine cut is not more than the total length of segments on the line, receiving projection from both sides. For simplicity of description, let us call by *horizontal (vertical) 1-dark point* a point receiving horizontal (vertical) projection from both sides. Then, for a horizontal (vertical) line, the set of vertical (horizontal) 1-dark points form the segment adding which would make the line become a 1-guillotine cut.

Lemma 3.2 *Let H (V) be the set of all horizontal (vertical) 1-dark points. Then there exists either a horizontal line L such that*

$$length(L \cap H) \le length(L \cap V)$$

or a vertical line L such that

$$length(L \cap H) \ge length(L \cap V).$$

Proof. First, assume that the area of H is not smaller than the area of V. Denote $L_a = \{(x, y) \mid x = a\}$. Then areas of H and V can be represented by

$$\int_{-\infty}^{+\infty} length(L_a \cap H)da$$

and

$$\int_{-\infty}^{+\infty} length(L_a \cap V)da,$$

respectively. Since

$$\int_{-\infty}^{+\infty} length(L_a \cap H)da \geq \int_{-\infty}^{+\infty} length(L_a \cap V)da,$$

there must exist a such that

$$length(L_a \cap H) \geq length(L_a \cap V).$$

Similarly, if the area of H is smaller than the area of V, then there exists a horizontal line L such that

$$length(L \cap H) \leq length(L \cap V).$$

□

By Lemma 3.2, without loss of generality, we may assume that there exists a horizontal line L such that

$$length(L \cap H) \leq length(L \cap V),$$

that is,

$$length(L \cap H) \leq proj_x(P_1) + proj_x(P_2) - proj_x(P)$$

where P_1 and P_2 are subpartitions obtained from P by the line which becomes a 1-guillotine cut after adding segment $L \cap H$ to the partition P. By induction hypothesis,

$$guil(P_i) \leq 2 \cdot length(P_i) - proj_x(P_i) - proj_y(P_i)$$

for $i = 1, 2$. Note that

$$proj_y(P) \quad \leq \quad proj_y(P_1) + proj_y(P_2).$$

Therefore,

$$\begin{aligned} guil(P) &= guil(P_1) + guil(P_2) + length(L \cap H), \\ &\leq 2\sum_{i=1}^{2} length(P_i) - \sum_{i=1}^{2} proj_x(P_i) - \sum_{i=1}^{2} proj_y(P_i) + length(L \cap H) \\ &\leq 2 \cdot length(P) - proj_x(P) - proj_y(P). \end{aligned}$$

□

Mitchell [17, 18] used a different way to present the proof of Theorem 3.1. He symmetrically charged a half of the length of added segment to those parts of segments in P which face to 1-dark points. Since charge must be performed symmetrically, each point in P can be charged at most twice during the entire modification from a rectangular partition to a 1-guillotine rectangular partition. Therefore, the total length of added segments is at most $length(P)$ and hence Theorem 3.1 holds. Actually, this argument is equivalent to the current proof of Theorem 3.1. In fact, only projections from both sides exist (Case 2), $proj_x(P)$ or $proj_y(P)$ can contribute something against the length of the added segment.

4 m-Guillotine Cut

Mitchell [19] extended the 1-guillotine cut to the m-guillotine cut in the following way: A point p is a horizontal (vertical) *m-dark* point if the horizontal (vertical) line passing through p intersects at least $2m$ vertical (horizontal) segments of the considered rectangular partition P, among which at least m are on the left of p (above p) and at least m are on the right of p (below p). Let H_m (V_m) denote the set of all horizontal (vertical) m-dark points. An *m-guillotine cut* is either a horizontal line L satisfying

$$L \cap H_m \subseteq L \cap P$$

or a vertical line L satisfying

$$L \cap V_m \subseteq L \cap P.$$

A rectangular partition is m-guillotine if it can be realized by a sequence of m-guillotine cuts. The minimum m-guillotine rectangular partition can also be computed by dynamic programming in $O(n^{10m+6})$ time. In fact, at each step, an m-guillotine cut has at most $O(n^{2(m+1)})$ choices. There are $O(n^4)$ possible rectangles appearing in the algorithm. Each rectangle has $O(n^{8m})$ possible boundary conditions. By a similar argument, Mitchell [19] established the following result.

Theorem 4.1 *For any rectangular partition P, there exists an m-guillotine rectangular partition P' covering P such that*

$$length(P') \leq (1 + \frac{1}{m})length(P).$$

Corollary 4.2 *There exists a $(1+\varepsilon)$-approximation with running time $n^{O(\log 1/\varepsilon)}$ for MELRP.*

From the 1-guillotine cut to the m-guillotine cut, it has no technical difficulty. But, why Mitchell did not do such an extention until Arara [1] published his remarkable results we mentioned at the beginning of this article? The answer is that before Arora's breakthrough, nobody was thinking in this way. Indeed, the importance of Arora's work [1] is more on opening people's mind than proposing new techniques. In terms of techniques, m-guillotine is more powerful. For example, we do not know how to apply Arora's techniques in [1] to obtain polynomial-time approximation schemes for the MELRP, the rectilinear Steiner arborescence problem [16], and the symmetric Steiner arborescence problem [5]. But, the m-guillotine cut works for them. For problems in high dimensional space, in particular, for geometric optimization problems in three or more dimensional space, Arora [1] provided $(1 + \varepsilon)$-approximation with running time $n^{O(\log 1/\varepsilon(\log n)^{d-1})}$. But, the m-guillotine cut can still provide with polynomial-time approximation schemes. We will give more explanation in the next section.

5 Portals

Arora's polynomial-time approximation scheme in [1] is also based on a sequence of cuts on rectangles. For example, let us consider Euclidean SMT. Initially, use a square to cover n

input points. Then with a tree structure, partition this square into small rectangles each of which contains one given point. By choosing cut line in a range between 1/3 and 2/3 of an edge, Arora managed the tree structure to have depth $O(\log n)$.

To reduce the number of crosspoints at each cut line, Arora [1] use a different technique. This technique is the portal. The portals are points on cut line equally dividing cut segments. For Euclidean SMT (or Euclidean TSP, etc), crosspoints of the Steiner tree on a cut line can be moved to portals. This would reduce the number of crosspoints on the cut line. Suppose the number of portals is p. It can be proved that by properly choosing cut line, at each level of the tree structure moving crosspoints to portals would increase the length of the tour within three pth of the total length of the Steiner tree. Since the tree structure has depth $O(\log n)$, the total length of the resulting Steiner tree is within $(1+\frac{3}{p})^{O(\log n)}$ times the length of the optimal one. To obtain $(1+\frac{3}{p})^{O(\log n)} \leq 1+\varepsilon$, we have to choose $p = O(\frac{\log n}{\varepsilon})$.

For problems in 3 or higher-dimensional space, the cut line should be replaced by cut plane or hyperplane. The number of portals would be $O((\frac{\log n}{\varepsilon})^2)$ or more. With so many possible crosspoints, the dynamic programming cannot run in polynomial time. However, the m-guillotine cut has at most $2m$ crosspoints in each dimension and m is a constant with respect to n. Therefore, the polynomial-time for the dynamic programming would be preserved under increasing dimension.

Combining the two techniques (the portal and the m-guillotine cut) can reduce the running time for dynamic programming. In fact, the portal technique first reduces the number of possible positions for crosspoints to $O(\frac{\log n}{\varepsilon})$ and this enables us to choose $2m$ from the $O(\frac{\log n}{\varepsilon})$ positions to form a m-guillotine cut ($m = 1/\varepsilon$). Therefore, the dynamic programming for finding the best such partition runs in time $n^c(\log n)^{O(1/\varepsilon)})$ where c is a constant. This is the basic idea of Arora [2] and Mitchell [20]. Arora's work [2, 3] also contains a new technique about the tree structure of partition. Indeed, it is an earlier and better work compared with Mitchell [20].

The portal technique cannot apply to the MELRP, the rectilinear Steiner arborescence, and the symmetric rectilinear Steiner arborescence. In fact, for these three problems, moving crosspoints to portals is sometimes impossible. Therefore, it is an open problem whether there exists a polynomial-time approximation scheme with running time $O(n^c(\log n)^{O(1/\varepsilon)})$.

The power of the m-guillotine cut also has certain limitation. For example, we do not know how to establish a polynomial-time approximation scheme without including total length of given segments in the problem of interconnecting highways [4]. This provides some opportunities for further developments of those elegant techniques.

References

[1] S. Arora, Polynomial-time approximation schemes for Euclidean TSP and other geometric problems, *Proc. 37th IEEE Symp. on Foundations of Computer Science*, (1996) pp. 2-12.

[2] S. Arora, Nearly linear time approximation schemes for Euclidean TSP and other geometric problems, I*Proc. 38th IEEE Symp. on Foundations of Computer Science*, (1997) pp. 554-563.

[3] S. Arora, Polynomial-time approximation schemes for Euclidean TSP and other geometric problems, *Journal of the ACM* 45 (1998) 753-782.

[4] X. Cheng, J.-M. Kim and B. Lu, A polynomial time approximation scheme for the problem of interconnecting highways, to appear in *Journal of Combinatorial Optimization.*

[5] X. Cheng, B. DasGupta, and B. Lu, A polynomial time approximation scheme for the symmetric rectilinear Steiner arborescence problem, to appear in *Journal of Global Optimization.*

[6] D.Z. Du, F.K. Hwang, M.T. Shing and T. Witbold: Optimal routing trees, *IEEE Transactions on Circuits* 35 (1988) 1335-1337.

[7] D.-Z. Du, L.-Q. Pan, and M.-T. Shing, Minimum edge length guillotine rectangular partition, Technical Report 0241886, Math. Sci. Res. Inst., Univ. California, Berkeley, 1986.

[8] D.-Z. Du, D.F. Hsu, and K.-J Xu, Bounds on guillotine ratio, *Congressus Numerantium* 58 (1987) 313-318.

[9] D.-Z. Du, On heuristics for minimum length rectangular partitions, Technical Report, Math. Sci. Res. Inst., Univ. California, Berkeley, 1986.

[10] D.-Z. Du and Y.-J. Zhang: On heuristics for minimum length rectilinear partitions, *Algorithmica*, 5 (1990) 111-128.

[11] T. Gonzalez and S.Q. Zheng, Bounds for partitioning rectilinear polygons, *Proc. 1st Symp. on Computational Geometry*, 1985.

[12] T. Gonzalez and S.Q. Zheng, Improved bounds for rectangular and guillotine partitions, *Journal of Symbolic Computation* 7 (1989) 591-610.

[13] A. Lingas, R. Y. Pinter, R. L. Rivest, and A. Shamir, Minimum edge length partitioning of rectilinear polygons, *Proc. 20th Allerton Conf. on Comm. Control and Compt.*, Illinos, 1982.

[14] A. Lingas, Heuristics for minimum edge length rectangular partitions of rectilinear figures, *Proc. 6th GI-Conference*, Dortmund, January 1983 (Springer-Verlag).

[15] C. Levcopoulos, Fast heuristics for minimum length rectangular partitions of polygons, *Proc 2nd Symp. on Computational Geometry*, 1986.

[16] B. Lu and L. Ruan, Polynomial time approximation scheme for the rectilinear Steiner arborescence problem, *Journal of Combinatorial Optimization* 4 (2000) 357-363.

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

[18] J.S.B. Mitchell, A. Blum, P. Chalasani, S. Vempala, A constant-factor approximation algorithm for the geometric k-MST problem in the plane *SIAM J. Comput.* 28 (1999), no. 3, 771–781.

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

[20] J.S.B. Mitchell, Guillotine subdivisions approximate polygonal subdivisions: Part III - Faster polynomial-time approximation scheme for geometric network optimization, preprint, 1997.