Fields Institute Communications
Volume **37**, 2003

A Tale on Guillotine Cut

Mihaela Cardei
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
mihaela@cs.umn.edu

Xiuzhen Cheng
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
cheng@cs.umn.edu

Xiaoyan Cheng
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
xiaoyan@cs.umn.edu

Ding-Zhu Du
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
dzd@cs.umn.edu

Abstract. The guillotine cut is an important tool to design polynomial-time approximation schemes for geometric optimization problems. In this article, we survey its history and recent developments.

1 Guillotine Cut

Robespirre (1758-1794) introduced the guillotine cut in French revolution. Nowadays, the guillotine cut has become an important technique to design PTAS (polynomial-time approximation schemes) for geometric optimization problems.

Roughly speaking, a *guillotine cut* is a subdivision with a line which divides given area into at least two subarea. To make our expanation more meaningful, let us consider a specific problem.

The minimum edge-length rectangular partition (MELRP) was first proposed by Lingas, Pinter, Rivest, and Shamir [15]. It can be stated as follows: Given a rectilinear polygon possibly with some rectangular holes, partition it into rectangles with minimum total edge-length.

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

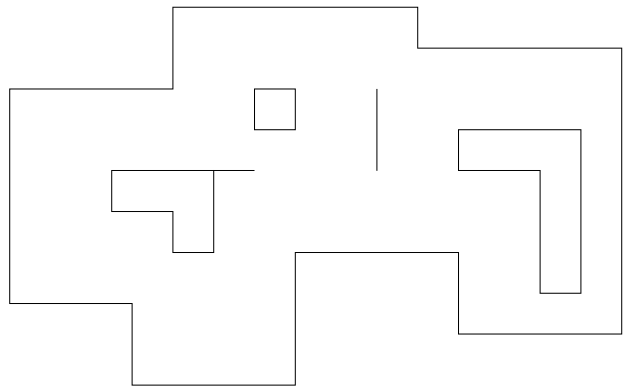

Figure 1 Rectilinear polygon with holes.

There are several applications mentioned in [15] 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.

Lemma 1.1 *There exists an optimal rectangular partition in which each maximal line-segment contains a vertex of the boundary.*

Proof Consider a minimum length rectangular partition P. Suppose P has a maximal vertical line-segment $[A, B]$ which does not contain any vertex of the boundary (see Fig. 2). Then two endpoints A and B must lie on the interior of two horizontal line-segments in P or the boundary. Suppose there are r horizontal segments touching the interior of $[A, B]$ from right and l horizontal segments touching the interior of $[A, B]$ from left. We claim that $r = l$. In fact, if $r > l$ (or $r < l$), then we can move $[A, B]$ to the right (or left) to reduce the total length of the rectangular partition, contradicting the minimality of P.

Since $r = l$, moving $[A, B]$ to either right or left does not increase the total length of P. Let us keep moving $[A, B]$ to the left. Then we must be able to move $[A, B]$ to contain a vertex of the boundary; otherwise, $[A, B]$ would be moved to overlap with another vertical segment in P, so that the total length of the rectangular partition is reduced, contradicting the optimality of P again. □

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

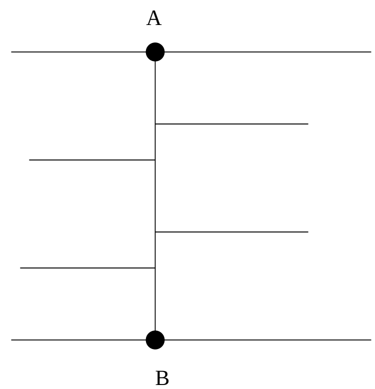

Figure 2 Maximal vertical line-segment.

case in $O(n^4)$ time. With this idea, Lingas [16] gave the first constant-bounded approximation; its performance ratio is 41. Later, Du [10, 11] improved the algorithm and obtained an approximation with performance ratio 9. Meanwhile, Levcopoulos [17] 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 [7] on application of dynamic programming to optimal routing trees, Du, Pan, and Shing [8] 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. The guillotine cut features dynamic programming since each guillotine cut breaks a minimum length guillotine rectangular partition problem into two or more subproblems.

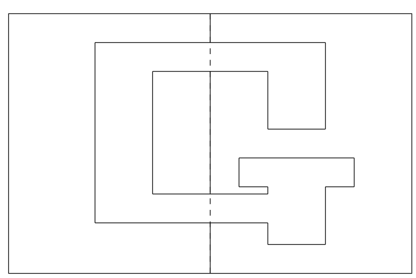

Figure 3 The guillotine cut features dynamic programming.

Moreover, Du *et al* [8] noticed that the minimum length guillotine rectangular partition also satisfies the property stated in Lemma 1.1. Hence, the minimum length guillotine rectangular partition 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 [12]) that the MELRP in this case is still NP-hard. Du *et al* [8] showed that the minimum length guillotine rectangular partition as an approximation for the MELRP has performance rato at most 2 in this special case. (This ratio is improved to 1.75 by Gonzalez and Zheng [13].) The following is a simple version of their proof, obtained by Du, Hsu, and Xu [9].

Theorem 1.2 *The minimum length guillotine rectangular partition is an approximation with performance ratio 2 for the MELRP.*

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

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

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

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

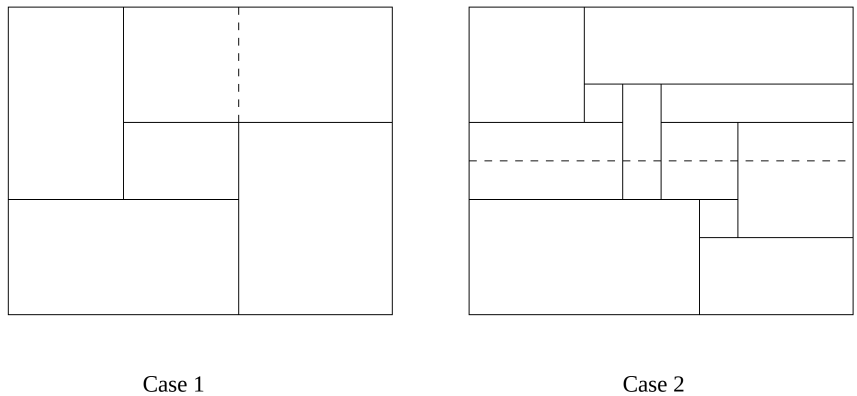

Figure 4 The proof of Theorem 1.2.

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,

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

□

Gonzalez and Zheng [13] improved the constant 2 in Theorem 1.2 to 1.75 with a very complicated case-by-case analysis. Du, Hsu, and Xu [9] also extended the idea of guillotine cut to the convex partition problem.

2 1-Dark Points

In the proof of Theorem 1.2, we may note that in case 2, every point on the cut line receives projection from two sides, both above and below. We call such a point as a vertical 1-dark point. Namely, a point in considered area is called a *vertical (horizontal) 1-dark point* if starting from the point along vetical (horizontal) line going either direction would meet at least one horizontal (vertical) segment in considered partition. Indeed, the term $proj_x$ takes advantage in the induction proof only on those vertical 1-dark point, since the cut line lies in the area of vertical 1-dark point. After cutting, the same size of term $proj_x$ would be kept in each of the two inequalities for subproblems. When the two inequalities are added together, the size of term $proj_x$ is doubled.

There is an alternative way to take the advantage of 1-dark points, which can be seen in the following alternative proof of Theorem 1.2.

Alternative Proof of Theorem 1.2.

Proof Consider a rectangular partition P.

Case 1. There exists a vertical segment s having length $\geq 0.5a$. Apply a guillotine cut along this segment s and charge 1 to the segment s.

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. Charge 0.5 to those horizontal segments, which directly face the cut. Note that every point on the cut is a vertical 1-dark point. Therefore, charged horizontal segments have a total length equal to exactly twice of the length of the cut.

Since each vertical segment in P is charged at most once and each horizontal segment is charged at most twice, the total length of added segments in guillotine cuts cannot exceed the total length of P. This completes the proof of Theorem 1.2. □

How do we find a guillotine cut consisting of 1-dark points? This is a central part of the argument in [8, 9]. Du et al [8] succeded in the special case, but were unable to extend their excellent idea to the general case.

3 1-Guillotine Cut and Mitchell's Lemma

"inspired by the proof in [8]" (quote from Mitchell [19, 20]), Mitchell made a significant progress in exploring the idea of guillotine cut.

First, he found a close relationship between 1-dark points and the guillotine cut by extending the guillotine cut to the 1-guillotine cut. A vertical (horizontal) cut is called *1-guillotine cut* if it consists of all vertical (horizontal) 1-dark points on the vertical (horizontal) line passing through the cut. (See Fig. 5.) This line will be called a *cut line*.

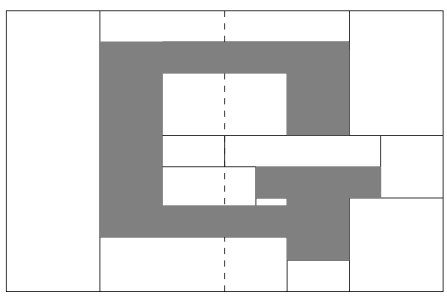

Figure 5 1-guillotine cut

Secondly, he found a very important relationship between vertical 1-dark points and horizontal 1-dark points.

Lemma 3.1 (Mitchell's Lemma) *Let H (V) be the set of all horizontal (vertical) 1-dark points. Then there exists either a horizontal line L such that*

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

or a vertical line L such that

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

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

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

□

This lemma actually means that there exists either a vertical 1-guillotine cut of length not exceeding the total length of segments consisting of all horizontal 1-dark points on the cut line, or a horizontal 1-guillotine cut of length not exceeding the total length of segments consisting of all vertical 1-dark points on the cut line. Namely, there always exists a 1-guillotine cut such that its length can be *symmetrically* charged to those segments parallel to the cut line, with value 0.5 to each side.

A rectangular partition is called a *1-guillotine rectangular partition* if it can be performed by a sequence of 1-guillotine cuts. It can be showed that there exists a minimum 1-guillotine rectangular partition such that every maximal segment contains at least a vertex of the boundary.

Now, the question is whether the 1-guillotine cut also features the dynamic programming. The answer is yes. In fact, the 1-guillotine cut partitions a rectangular partition problem into two subproblems with boundary conditions, since after a 1-guillotine cut, two open segments may be created on the boundary. This boundary condition increases the number of subproblems in the dynamic programming. Since each subproblem is based on a rectangle with four sides. The condition on each side can be described by two possible open segments at the two ends. Hence each side has $O(n^2)$ possible conditions. So, the total number of boundary conditions is $O(n^8)$. This gives that the total number of possible subproblems is $O(n^{12})$. For each problem, there are $O(n^3)$ possible 1-guillotine cuts. Therefore, the minimum 1-guillotine rectangular partition can be computed by a dynamic programming in $O(n^{15})$ time.

With 1-guillotine cuts, the approximation ratio 2 can be established not only for the special case, but also in general. First, we use a rectangle to cover the input rectangular polygon with holes. Then, we can cut the rectangle each time into two rectangles with the 1-guillone cut.

Theorem 3.2 *Every rectangular partition P can be modified into a 1-guillotine partition by adding some segments of total length not exceeding to the total length of P.*

Proof Each time, if a 1-guillotine cut already exists in segments belonging to P, then we use it to divide the considered rectangle into two parts. If such a 1-guillotine cut does not exist, then by Mitchell's lemma, there exists a 1-guillone cut whose length can be symmetrically charged to parallel segments in P. We add segments in this 1-guillotine cut into P. Since charge is performed symmetrically, no segment in P can be charged more than twice. Therefore, added segments have a total length not exceeding the total length of P. □

This theorem actually means that the minimum 1-guillotone rectangular partition is a polynomial-time approximation with performance 2 for minimum rectangular partition.

4 m-Guillotine Cut

Mitchell [21] 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). A horizontal (vertical) cut is an *m-guillotine cut* if it consists of all horizontal (vertical) m-dark points on the cut line. In other words, 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

$$L \cap V_m \subseteq L \cap P,$$

where P is the considered partition. A rectangular partition is m-guillotine if

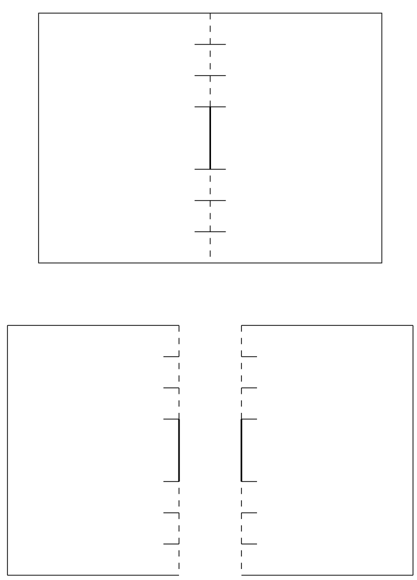

Figure 6 m-guillotine cut results in $2m$ open segments on each subproblem's boundary.

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 $O(n^{10m+5})$ time. In fact, at each step, an m-guillotine cut has at most $O(n^{2m+1})$ 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 [21] established the following result.

Lemma 4.1 (Mitchell's Lemma) *There exists either a horizontal line L such that*

$$length(L \cap H_m) \leq length(L \cap V_m)$$

or a vertical line L such that

$$length(L \cap H_m) \geq length(L \cap V_m).$$

Theorem 4.2 *Every rectangular partition P can be modified into an m-guillotine rectangular partition P' with total length*

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

Corollary 4.3 *There exists a polynomial-time $(1+\varepsilon)$-approximation with running time $n^{O(\log 1/\varepsilon)}$ for MELRP.*

From the 1-guillotine cut to the m-guillotine cut, there is no technical difficulty. Unfortunately, this extention was established just a few weeks after Arora [1] published his remarkable results.

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-MST (minimum spanning tree), k-TSP, and k-SMT, have polynomial-time approximation schemes. More precisely, for any $\varepsilon > 0$, there exist approximation algorithms for those problems, running in time $n^{O(1/\varepsilon)}$, which produce approximation solution within $1 + \varepsilon$ from optimal. It made Arora's research be reported in *New York Times*. Several weeks later, Mitchell [21] claimed that his earlier work [19] (its journal version [20]) 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 [22] claimed again that his approach can do a similar thing.

Next, let us study Arora's seminal work to find out its relationship with the guillotine cut.

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 rectilinear SMT. Initially, use a minimal square to cover n input points. Then with a tree structure, partition this square into small rectangles each of which contains one given point. Arora managed the tree structure to have depth $O(\log n)$ with the following techniques:

(1) Equally divide the initial square into $n^2 \times n^2$ lattice. Move each given point to its closest lattice point.

(2) Choose cut line in a range between 1/3 and 2/3 of a longer edge (or an edge for a square), through the middle between two adjacent lattice points.

The following lemmas explain these two techniques.

Lemma 5.1 *Let P be the set of n given points and P' the set of n lattice points closest to n given points, respectively. If there is a PTAS for P', then there exists a PTAS for P.*

Proof Let $T_\varepsilon(P')$ be a polynomial-time ε-approximation for rectilinear SMT on P'. That is,

$$length(T_\varepsilon(P')) \le (1+\varepsilon) length(RSMT(P')).$$

where $RSMT(P')$ is the rectilinear Steiner minimum tree on P'. Note that

$$|length(RSMT(P)) - length(RSMT(P'))| \le L/n$$

where L is the edge length of the initial square. Since the square is minimal, L is not bigger than the length of the minimum spanning tree on P and hence not bigger than $1.5 length(RSMT(P))$.

Construct a tree T interconnecting points in P from $T_{\varepsilon/2}(P')$ by connecting each point in P' to its corresponding point in P. Then

$$\begin{aligned} length(T) &\le length(T_{\varepsilon/2}(P')) + L/n \\ &\le (1+\varepsilon/2) \cdot length(RSMT(P') + L/n \\ &\le (1+\varepsilon/2)(length(RSMT(P) + L/n) + L/n \\ &= (1+\varepsilon/2) length(RSMT(P)) + (2+\varepsilon/2)L/n \\ &\le (1+\varepsilon/2 + (2+\varepsilon/2)\cdot 1.5/n) length(RSMT(P)). \end{aligned}$$

Note that for sufficiently large n,

$$(2+\varepsilon/2)\cdot 1.5/n < \varepsilon/2$$

that is,

$$length(T) < (1+\varepsilon) length(RSMT(P)).$$

□

Based on this lemma, we will work on P' instead of P.

Lemma 5.2 *With technique (2), the binary tree structure of the partition has* $O(\log n)$ *levels.*

Proof With technique (2), the rectangle at the ith level has area at most $L^2(2/3)^{i-1}$. Since the ratio between the lengths of longer edge and shorter edge is at most three, the rectangle at the last level, say the sth level, has area at least $(1/3)(L/n^2)^2$. Therefore, $L^2(2/3)^{s-1} \ge (1/3)(L/n^2)^2$. That is, $s = O(\log n)$. □

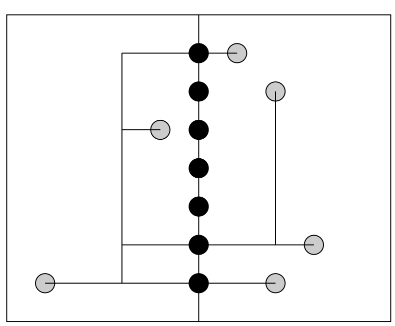

Figure 7 Portals

To reduce the number of crosspoints at each cut line, Arora [1] used a different technique. This technique is the portal. Portals are points on cut line equally dividing cut segments. For rectilinear SMT (or Euclidean SMT, 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. The following lemma shows that by properly choosing cut line, at each level of the tree structure moving crosspoints to portals would increase the length of the tree within a certain amount.

Lemma 5.3 *By properly choosing cut line, at each level of the tree structure moving crosspoints to portals would increase the length of the tree within three pth of the total length of the Steiner tree.*

Proof Consider each rectangle R at a certain level. Suppose its longer edge has length a and shorter edge has length b ($b \leq a$). Look at every possible cut in a range between 1/3 and 2/3 of a longer edge. Choose the cut line to intersect the Steiner tree with the smallest number of points, say c points. Then the length of the part of the Steiner tree lying in rectangle R is at least $ca/3$. Moving c crosspoints to portals requires to add some segments of total length at most $cb/(p+1) \leq ca/(p+1) \leq (3/p)(ca/3)$. □

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 $p = O(\frac{\log n}{\varepsilon})$.

Summerizing the above, we already proved the structure theorem of Arora.

Theorem 5.4 (Structure Theorem) *For any RSMT T^*, there exists a $(1+\varepsilon)$-approximation tree T which can be constructed with $(1/3, 2/3)$-partition and p portals on each cut where $p = O(\frac{\log n}{\varepsilon})$. Moreover, the tree structure of the $(1/3, 2/3)$-partition has $O(\log n)$ levels.*

Now, we describe how to find such a $(1+\varepsilon)$-approximation in the structure theorem. We employ dynamic programming to find the shortest one among the trees with the same structure as the ε-approximation.

To estimate the running time of dynamic programming, we first note that each subproblem is characterized by a rectangle and conditions on the boundary. There are $O(n^8)$ possible rectangles. Each rectangle has four sides. One of them must contain p portals. However, each of other three may contain less than p portals resulting from previous cuts. Thus, the number of positions for portals on each of these three sides is $O(n^4)$. Hence, the total number of portal positions on the boudary is $O(n^{20})$. For each fixed set of portal possitions, we need also consider whether a portal is a crosspoint or not and how crosspoints are connected to each other inside the rectangle. It brings us $2^{O(p)}$ possibilities. Therefore, the total number of possible subproblems is $n^{O(1/\varepsilon)}$.

Moreover, in each iteration of dynamic programming, the number of all possible cuts is $O(n^2)$. Therefore, the dynamic programming runs in time $n^{O(1/\varepsilon)}$.

6 m-Guillotine Cuts with Portals

Let us first compare the m-guillotine cut with the portal.

For problems in three 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.

The portal technique cannot be applied to the MELRP, the rectilinear Steiner arborescence[18], and the symmetric rectilinear Steiner arborescence [6]. In fact, for these three problems, moving crosspoints to portals is sometimes impossible. But, the m-guillotine cut works well in these problems.

In the other hand, the m-guillotine cut cannot be applied to grade Steiner tree [14], Euclidean k-median, and Euclidean facility location [4]. In fact, the m-guillotine cut may change the topologic structure of the connection, which would change the cost of connection lines. However, the portal technique can be successfully used for those problems.

Both techniques can be applied to the rectilinear SMT problem. However, if we count the runing time carefully, then it is not very hard to see that the dynamic programming with the m-guillotine cut takes less time than that with the portal. In fact, the m-guillotine cut allows us to reduce the number of crosspoints on a cut to a constant $O(1/\varepsilon)$ while the portal technique can only reduce the number to $O((\log n)/\varepsilon)$.

It is true that the m-guillotine cut has several advantages compared with than the portal technique. But, why Mitchell did not do such an extention from the 1-guillotine cut to the m-guillotine cut before Arara [1] published his remarkable results? 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.

Now, we discuss how to combine the m-guillotine cut with the portal. This combination would 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 [22]. Arora's work [2, 3] also contains a new

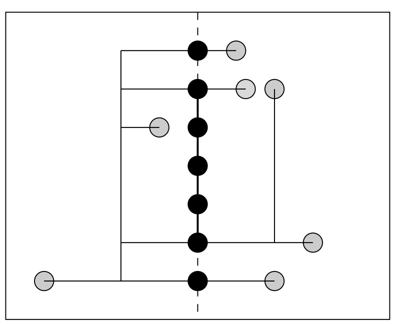

Figure 8 m-guillotine cut with portals

technique about the tree structure of partition. Indeed, it is an earlier and better work compared with Mitchell [22].

It is an open problem whether the MELRP, the rectilinear Steiner arborescence, the symmetric rectilinear Steiner arborescence, grade Steiner tree, Euclidean k-median, and Euclidean facility location have a PTAS with running time $n^c(\log n)^{O(1/\varepsilon)}$.

The power of the m-guillotine cut and the portal 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 [5]. This provides another opportunity for further development of these elegant techniques. Therefore, it is an open problem whether there exists a PTAS for the problem of interconnecting highways.

There are some geometric problems which both the portal and the m-guillotine cut cannot be applied to, such as rectilinear SMT with obstructions. Those problems encourage us to find new techniques.

References

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

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

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

[4] Arora, S., Raghavan, P., and Rao, S. *Approximation Schemes for Euclidean k-medians and related problems*, ACM STOC (1998), 106–113.

[5] Cheng, X., Kim, J.-M., and Lu, B. *A polynomial time approximation scheme for the problem of interconnecting highways*, Journal of Combinatorial Optimization **5(3)** (2001), 327–343.

[6] Cheng, X., DasGupta, B., and Lu, B. *A polynomial time approximation scheme for the symmetric rectilinear Steiner arborescence problem*, Journal of Global Optimization **21** (2001), 385–396.

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

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

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

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

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

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

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

[14] Kim, J., Cardei, M., Cardei, I., and Du, D.-Z. *Polynomial-time approximation scheme for grade Steiner minimum trees*, manuscript (2002).

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

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

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

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

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

[20] Mitchell, J.S.B., Blum, A., Chalasani, P., and Vempala, S. *A constant-factor approximation algorithm for the geometric k-MST problem in the plane* SIAM J. Comput. **28(3)** (1999), 771–781.

[21] Mitchell, J.S.B. *Guillotine subdivisions approximate polygonal subdivisions: Part II - A simple polynomial-time approximation scheme for geometric k-MST, TSP, and related problem*, SIAM J. Comput. **292)** (1999), 515–544.

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