IMPROVED BOUNDS FOR RECTANGULAR AND GUILLOTINE PARTITIONS‡

TEOFILO F. GONZALEZ and SI-QING ZHENG
Department of Computer Science
University of California
Santa Barbara, CA 93106

ABSTRACT

We study the problem of partitioning a rectangle R with a set of interior points Q into rectangles by introducing a set of line segments of least total length. The set of partitioning line segments must include every point in Q. Since this problem is computationally intractable (NP-hard), several approximation algorithms for its solution have been developed. In this paper we show that the length of an optimal guillotine partition is not greater than 1.75 the length of an optimal rectangular partition. Since an optimal guillotine partition can be obtained quickly, we have an efficient approximation algorithm for finding near-optimal rectangular partitions.

INTRODUCTION

Given a rectangular boundary S and a set Q of points inside S, we study the problem of partitioning S into rectangles in such a way that every point in Q lies on at least one of the partitioning line segments and the total length of the partitioning line segments is least possible. Such a partition is called an ***optimal rectangular partition.*** Lingas ***et. al.*** [8] show that finding an optimal rectangular partition is a computationally intractable problem (NP-hard). Since then several approximation algorithms have been proposed, i.e., algorithms for which LEN(E_{apx}(I)) $\leq$ c * LEN(E_{opt}(I)), where E_{apx}(I) is the set of partitioning line segments given by the approximation algorithm, E_{opt}(I) is the set of partitioning line segments in an optimal solution, c is some constant, and LEN(E(I)) is the sum of the length of the partitioning line segments in E(I). Gonzalez and Zheng [4] present a divide-and-conquer approximation algorithm that generates solutions such that LEN(E_{apx}(I)) $\leq 3+\sqrt{3}$ * LEN(E_{opt}(I)). The time complexity for their algorithm is O(n^2), where n is the number of points in set Q. Recently, Levcopoulos [10] showed that it is possible to implement this approximation algorithm in O(n log n) time. In [5] an (n^4) approximation algorithm that guarantees solutions such that LEN(E_{apx}(I)) $\leq$ 3 * LEN(E_{opt}(I)). The approximation bound is smaller than the one in [4], however there is a substantial difference between the time complexity of these two algorithms. The algorithm in [5] consists of two steps. In the first step, the original problem is transformed into a simpler polynomial time solvable optimization problem; in the second step an existing O(n^4) algorithm is employed to solve the new optimization problem. In this paper we present an approximation algorithm that generates solutions whose objective function value is within 75% of the objective function value of an optimal solution. Gonzalez and Zheng [5] show how to modify the algorithm to generate approximation solutions to a more general version of the problem (bounded boundary problem). The approximation bound obtained for the new problem in [5] is smaller than the one in [7]. If instead of a rectangle we start with a rectilinear polygon and if instead of interior points, the polygon contains holes (a hole is a rectilinear polygon without interior holes), the problem of finding a good rectangular partition becomes much more complex. This problem has applications in computer-aided design of integrated circuits and system for dividing routing regions into channels [11]. Several approximation algorithms for this problem exist (see [1], [7], [9] and [10]). The algorithms with the smallest worst case approximation bound are the ones given in [9] and [10]. The algorithm given in [10] uses as a subalgorithm the procedure given in [4]. Since the approximation algorithm given in this paper generates solutions of the same form as the ones in [4] but closer to the optimal solution value, we conjecture that a smaller approximation bound can be obtained by using the algorithm in this paper.

‡ This research was supported in part by the National Science Foundation under Grant DCR - 8503163

We say that the rectangular partition E for I = (S,Q) has a *guillotine cut* if there is a line segment in E that partitions S into two rectangles. We say that a rectangular partition E for I = (S,Q) is a *guillotine partition* if either E is empty (note that Q must be empty) or E has a guillotine cut that partitions S into S_1 and S_2, and both E_1 (partitioning line segments from E in S_1) and E_2 (partitioning line segments from E in S_2) are guillotine partitions for $I_1 = (S_1,Q_1)$ and $I_2 = (S_2,Q_2)$, respectively. An *optimal guillotine partition* is a guillotine partition whose partitioning line segments have least total length. It is simple to see that any guillotine partition is a rectangular partition, but the converse is not true. An optimal guillotine partition can be found in O(n^5) time [12]. Du *et. al.* [2] show that the length of an optimal guillotine partition is no more than twice the length of an optimal rectangular partition. Therefore, an approximation algorithm for the rectangular partition problem is reduced to the problem of finding an optimal guillotine partition. Gonzalez, Shing and Zheng [3] present a simple proof for this fact and point out that it is unlikely that the dynamic programming algorithm can be speeded up. The algorithm in [4] always generates a guillotine partition, however this in not true for the algorithm given in [5].

For problem instance I = (S, Q), let E_{ogp}(I) be the set of partitioning line segments in an optimal guillotine partition and let E_{opt}(I) be the set of partitioning line segments in any optimal rectangular partition. In what follows we show that LEN(E_{ogp}(I)) $\leq$ 1.75 * LEN(E_{opt}(I)). Therefore, we have an O(n^5) approximation algorithm for the rectangular partitioning problem such that LEN(E_{apx}(I)) $\leq$ 1.75 * LEN(E_{opt}(I)).

BOUNDS FOR GUILLOTINE PARTITIONS

We use P to denote the tuple (I = (S , Q) , E(I)), where I is a problem instance and E(I) is any rectangular partition for I. We present a transformation that introduces a set of line segments E'(I) such that E'(I) $\cup$ E(I) forms a guillotine partition (of course E(I) $\cap$ E'(I) = $\emptyset$). The transformation is performed in such a way that LEN(E'(I) $\cup$ E(I)) $\leq$ 1.75 * LEN(E(I)). Applying this transformation to any optimal rectangular partition E_{opt}(I), we know that the resulting guillotine partition E'(I) $\cup$ E_{opt}(I) is such that LEN(E'(I) $\cup$ E_{opt}(I)) $\geq$ LEN(E_{ogp}(I)) since E_{ogp}(I) is an optimal guillotine partition.

Let E_v(I) and E_h(I) represent the sets of vertical and horizontal line segments in E(I), respectively. For a vertical (horizontal) line segment l, we use x(l) (y(l)) to denote the x-coordinate (y-coordinate) of l and use B(l) (L(l)) and T(l) (R(l)) to denote the y-coordinate (x-coordinate) of the lower (left) end point and the upper (right) end point of the vertical (horizontal) line segment l, respectively. The y-coordinate of the bottom and top side of S is given by B(S) and T(S), respectively. The x-coordinate of the left and right side of S is given by L(S) and R(S), respectively. Let X = R(S) - L(S) and let Y = T(S) - B(S).

Since rotation of P by 90 degrees generates an equivalent problem, we may assume without loss of generality that LEN(E_v(I)) $\leq$ LEN(E_h(I)). In what follows we claim that our transformation process introduces a set of vertical line segments E_v'(I) such that LEN(E_v'(I)) $\leq$ LEN(E_v(I)), and a set of horizontal line segments E_h'(I) such that LEN(E_h'(I)) $\leq$ 0.5 * LEN(E_h(I)). Therefore, LEN(E'(I) $\cup$ E(I)) = LEN(E_h'(I)) + LEN(E_v'(I)) + LEN(E(I)) $\leq$ 0.5 * LEN(E_h(I)) + LEN(E_v(I)) + LEN(E(I)) $\leq$ 0.5 * LEN(E_v(I)) + 1.5 * LEN(E(I)) $\leq$ 1.75 * LEN(E(I)).

We say that line segment l is *covered* by line segment l' if every point in l is in l'. The line segment l is said to be covered by E(I) if there is a line segment l' in E(I) such that l is covered by l'. We use the (corrupted) notation $l \in l'$ and $l \in$ E(I) to indicate line covering. The *overlap* of line segments l and l' is defined as the line segment $l \cap l'$. The overlap of two sets of line segments is defined similarly. We say that a line segment $l \in$ E(I) is a vertical (horizontal) *guillotine cut* of S if T(l) = T(S) (L(l) = L(S)) and B(l) = B(S) (R(l) = R(S)). Note that this definition is equivalent to the one for guillotine cuts introduced in the previous section. A line segment l is a vertical (horizontal) *full cut* of S if T(l) = T(S) (L(l) = L(S)) and B(l) = B(S) (R(l) = R(S)). Note that every guillotine cut is a full cut, but not every full cut is a guillotine cut, simply because a full cut is not required to be in E(I). Clearly, when there is a guillotine cut l of

S in E(I), P is partitioned into P_1 and P_2 without introducing any new line segment. At this point we (recursively) transform $E_1(I_1)$ and $E_2(I_2)$. If at each step of this recursive transformation we encounter an instance with a guillotine cut, then $E_v'(I) = E_h'(I) = \emptyset$ and our claim for the 1.75 bound follows. However, when there is no guillotine cut of S in E(I) we must introduce a full cut. Selecting the full cut is the crucial part of the transformation.

When there is no guillotine cut of S in E(I) we either introduce a vertical full cut or a set of vertical and horizontal full cuts, depending on the configuration of E(I). The concept of separability, as we shall see later, plays an important role in this decision. We say that a vertical full cut l is *left (right) covered by $E_v(I)$* if for every point p in l there exists a line segment $l' \in E_v(I)$ such that $x(l') \leq x(l)$ ($x(l') \geq x(l)$) and $B(l') \leq y(p) \leq T(l')$. A ***vertical through cut*** is a vertical full cut that is both left and right covered by $E_v(I)$. We say P is ***vertically separable*** if there is at least one vertical through cut in P.

At this time it is convenient to view our transformation as follows:

case
:E(I) = $\emptyset$: **return**;
:E(I) has a guillotine cut:
partition I along a guillotine cut and recursively transform the resulting subproblems;
:E(I) has no guillotine cut but it is vertically separable:
partition I along a vertical through cut that overlaps with at least one vertical line segment in E(I) and recursively transform the resulting subproblems;
endcase

If at each step of the transformation process just described we encounter an instance with no internal line segments or an instance with a guillotine cut or an instance that it is vertically separable, then since the transformation process introduces no new horizontal line segments, we know that $E_h'(I) = \emptyset$. The set $E_v'(I) \neq \emptyset$ if in the recursive process we encounter a nonempty problem instance without a guillotine cut. In lemma 1 we prove that for this case LEN($E_v'(I)$) $\leq$ LEN($E_v(I)$). We use the projection function, p($E_v(I)$), to project onto the right side of S the set of vertical line segments $E_v(I)$. The length of the projection is given by LEN(p(E_v(I))). Clearly, for every $E_v(I)$, LEN(p(E_v(I))) ≥ 0. Instead of proving that LEN(E_v'(I)) $\leq$ LEN(E_v(I)), we prove a stronger result, i.e., LEN(E_v'(I)) + LEN(p(E_v(I))) $\leq$ LEN(E_v(I)).

Lemma 1: For every P = (I,E(I)) our transformation process introduces a set of line segments $E_v'(I)$ such that, LEN(E_v'(I)) + LEN(p(E_v(I))) $\leq$ LEN(E_v(I)).
Proof: For brevity the proof is omitted. The proof appears in [6].

□

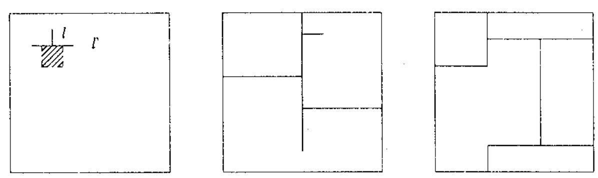
For problem instances P with the properties mentioned above, we know that our claim for the 1.75 bound is correct. For any general problem instance P we cannot yet claim this bound. This is because our transformation process is not complete, there are nonempty and not separable problem instances without a guillotine cut. For those cases we shall apply a three-phase transformation to be carried out by procedure HVH_CUT. In the first phase of procedure HVH_CUT, we introduce a set of horizontal full cuts to partition P into a set of vertically separable subinstances. Let H(I) be the set of horizontal full cuts introduced in phase one. Let $H_1(I)$ = H(I) $\cap$ $E_h(I)$ and let $H_1'(I)$ = H(I) - $H_1(I)$. At this point it is impossible to prove that LEN(H_1'(I)) $\leq$ 0.5 LEN(H_1(I)). This is why we need to perform the following steps. In the second phase, each of the vertically separable problem instances constructed in phase one is partitioned by introducing a vertical through cut. The vertical through cuts are carefully selected so that in the next phase we can find a set of horizontal guillotine cuts. Let $H_3(I)$ be this set of horizontal guillotine cuts. Note that $H_3(I) \subseteq E_h(I)$. At this point we claim that LEN(H_1'(I)) $\leq$ 0.5(LEN(H_1(I)) + LEN(H_3(I))). Our transformation process is formally defined below.

procedure TRANS(P = (I = (S , Q) , E(I)))
  case
    :E(I) is empty: **return**;
    :P has a guillotine cut l:
      partition P (along l) into P_1 and P_2;
      recursively apply TRANS to P_1 and P_2;
    :P is vertically separable:
      let l be any vertical through cut in P that overlaps with at least one vertical
        line segment in E(I);
      partition P (along l) into P_1 and P_2;
      recursively apply TRANS to P_1 and P_2;
    :**else**: use procedure HVH_CUT to partition P into P_1, ..., P_u;
      recursively apply TRANS to each P_i;
  endcase
end of procedure TRANS

From procedure TRANS and our informal description of procedure HVH_CUT, we know that every vertical line segment introduced is a vertical through cut. Therefore, a proof similar to the one for lemma 1 can be used to show that LEN(E_v'(I)) $\leq$ LEN(E_v(I)). To prove our 1.75 bound it is only required to show that for every P we invoke procedure HVH_CUT, LEN(H_1'(I)) ≤ 0.5 * (LEN(H_1(I)) + LEN(H_3(I))). Hereafter we concentrate on nonempty problem instances, P, which do not have a guillotine cut and are not separable.

The proof for the above bound is not simple. Before proving it we need to introduce some additional notation and prove some intermediate results.

We say that P' = (I' = (S', Q') , E(I')) $\subseteq$ P = (I = (S, Q) , E(I)) if T(S') $\leq$ T(S), B(S') $\geq$ B(S), L(S') $\geq$ L(S), R(S') $\leq$ R(S), Q' contains all the points in Q located inside (not in the boundary) S', and E(I') contains all the line segments in E(I) located inside S'. We say that P' = (I' = (S',Q') , E(I')) is empty if E(I') is empty, i.e., there are no line segments in E(I'). An important property of empty subproblems is given by the following lemma.



(a) loose ends (b) dangling corners (c)
Figure 2.1:

Lemma 2: If there is an empty subproblem P' = (I',E(I')) $\subseteq$ P = (I,E(I)) and there is a line segment $l \in E_v$(I) such that L(S') < x(l) < R(S') and B(l) = T(S') (T(l) = B(S')), then there is a horizontal line segment $l' \in E_h$(I) such that y(l') = T(S') (y(l') = B(S')), L(l') $\leq$ L(S') and R(l') $\geq$ R(S').
Proof: For brevity the proof is omitted. The proof appears in [6].

□

Among all vertical through cuts in a vertically separable problem P the one with smallest x-coordinate and the one with largest x-coordinate are referred to as the *leftmost vertical through cut lm(P)* and the *rightmost vertical through cut rm(P),* respectively. Note that for some P, the leftmost vertical cut could also be the rightmost vertical cut. In what follows we define some separable subproblems (via procedure MARK), then examine some of their properties and finally show how to use these subproblems and their properties to perform the three phases in procedure HVH_CUT.

Let $y_1 < y_2 <, ..., < y_s$ be the distinct y-coordinates of the set of line segments in $E_h(\mathrm{I})$. Let y_0 = B(S) and y_{s+1} = T(S). For $0 \leq t \leq u \leq s+1$, let $S_{t,u}$ denote the rectangle defined by ((L(S), y_t), (L(S), y_u), (R(S), y_u), (R(S), y_t)). Similarly, let $P_{t,u}$ denote P restricted to $S_{t,u}$. It is easy to see that if $P_{t,u}$ and $P_{u,v}$ are vertically separable but $P_{t,v}$ is not vertically separable, then either x(lm($P_{t,u}$)) > x(rm($P_{u,v}$)) or x(rm($P_{t,u}$)) < x(lm($P_{u,v}$)). In the former case we call the $P_{t,v}$ an *LR-decreasing problem* and in the latter case $P_{t,v}$ is referred to as an *LR-increasing problem*. If $P_{i,j}$ is separable then $P_{h,g}$, where $i \leq h \leq g \leq j$, is also separable. Furthermore x(lm($P_{h,g}$)) $\leq$ x(lm($P_{i,j}$)) and x(rm($P_{h,g}$)) $\geq$ x(rm($P_{i,j}$)). Note that $P_{i,i}$ is separable for all i; and for problem instances without guillotine cuts, $P_{i,i+1}$ is always separable.

Procedure MARK

```
j ← 0;
for i ← 1 to s+1 do
  if P_{j,i} is not separable then { assign the label A to y_{i-1}; j ← i - 1 }
endfor
i ← s+1;
for j ← s to 0 by -1 do
  if P_{j,i} is not separable then { assign the label B to y_{j+1}; i ← j + 1 }
endfor
```

end of MARK

Let $y_{i(1)} < y_{i(2)} < ... < y_{i(k)}$, be the y-coordinates labeled A and let $y_{j(1)} < y_{j(2)} < ... < y_{j(k')}$, be the y-coordinates labeled B by algorithm MARK. Note that the $y_{j()}$'s appear in reverse order from the way procedure MARK labels them. For convenience, let $y_{i(0)} = y_{j(0)}$ = B(S) and $y_{i(k+1)} = y_{j(k'+1)}$ = T(S). Since the $P_{i(p),i(p+1)}$ is separable and the algorithm selects the $y_{j()}$'s to represent maximal separable subproblems with respect to the previous $y_{j()}$'s, we know that it is impossible for two $y_{j()}$'s to be in the interval ($y_{i(p)}$, $y_{i(p+1)}$]. Similarly, it is impossible for two $y_{i()}$'s to be in the interval [$y_{j(p)}$, $y_{j(p+1)}$). Hence, k = k' and $y_{j(0)} = y_{i(0)} < y_{j(1)} \leq y_{i(1)} < y_{j(2)} \leq y_{i(2)} < ... < y_{j(k)} \leq y_{i(k)} < y_{j(k+1)} = y_{i(k+1)}$. In lemma 3 we prove an important property of LR-decreasing and LR-increasing problems.

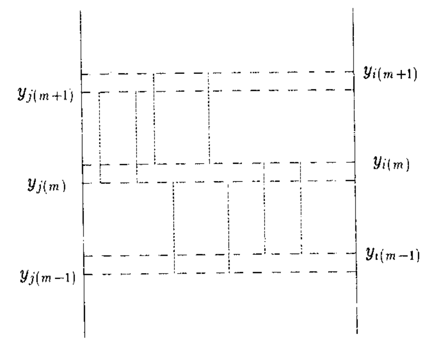


Figure 2.2:

Lemma 3: If $P_{i(m-1),i(m+1)}$ is an LR-decreasing (LR-increasing) problem then $P_{j(m-1),j(m+1)}$ is also an LR-decreasing (LR-increasing) problem.
Proof: For brevity the proof is omitted. The proof appears in [6].

□

Let LEFT = { l | $l \in E_h(\mathrm{I})$ and L(l) = L(S)} and RIGHT = { l | $l \in E_h(\mathrm{I})$ and R(l) =

R(S)}. In the following lemma we show that for each $y_{i()}$ and $y_{j()}$, there is a distinct horizontal line segment from LEFT or RIGHT associated with it. For $x_1 < x_2$ we use *HLS*(y,x_1,x_2) to represent the horizontal line segment with end points (x_1,y) and (x_2,y).

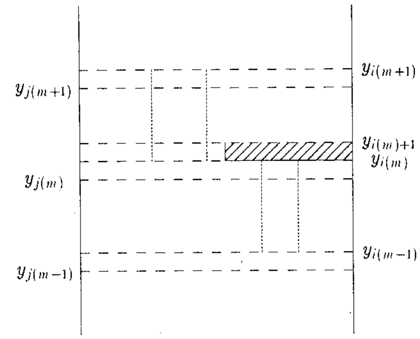


Figure 2.3:

Lemma 4:

(i) If $P_{i(m-1),i(m+1)}$ is an LR-decreasing (LR-increasing) problem then the line segment

l = HLS($y_{i(m)}$, x(rm($P_{i(m),i(m)+1}$)), R(S)) (HLS($y_{i(m)}$, L(S), x(lm($P_{i(m),i(m)+1}$)))) is in E_h(I);

and

(ii) if $P_{j(m-1),j(m+1)}$ is an LR-decreasing (LR-increasing) problem then then the line segment

l = HLS($y_{j(m)}$, L(S), x(lm($P_{j(m)-1,j(m)}$))) (HLS($y_{j(m)}$, x(rm($P_{j(m)-1,j(m)}$)), R(S))) is in E_h(I).

Proof: For brevity the proof is omitted. The proof appears in [6].

□

The line segment identified by lemma 4(i) is referred to by $l_{i(m)}$ and the one identified by lemma 4(ii) is referred to by $l_{j(m)}$. From lemma 3 and 4 we know that $l_{i(m)} \in$ LEFT (RIGHT) iff $l_{j(m)} \in$ RIGHT (LEFT). If y($l_{i(m)}$) = y($l_{j(m)}$), then it is not possible for $l_{i(m)}$ and $l_{j(m)}$ to overlap because we are assuming there are no horizontal guillotine cuts. Therefore, for each $y_{i()}$ and $y_{j()}$, there is a distinct horizontal line segment in LEFT or RIGHT associated with it. In the next two lemmas we show that each of the line segments identified by the previous lemma can be associated with a distinct line segment in E_h(I) such that their total length is at least X. This is an important property needed to establish our 1.75 bounds. In lemma 5 we show that for each $l_{i(m)}$ and $l_{j(m)}$ there is another line segment in E_h(I) such that the sum of their length is at least X. Since this does not necessarily guarantee a 1-1 association between line segments, we need lemma 6.

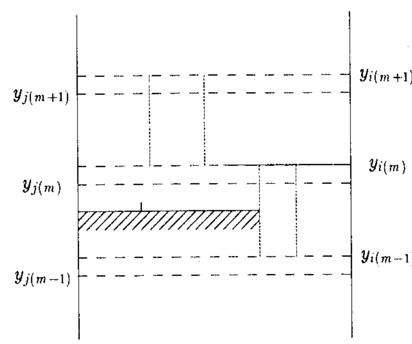


Figure 2.4:

Lemma 5:
(i) If $P_{i(m-1),i(m+1)}$ is an LR-decreasing (LR-increasing) problem then there exists at least one line segment l in E_h(I) such that l = HLS(y, L(S), x(lm($P_{i(m-1),j(m)}$))) (HLS(y, x(rm($P_{i(m-1),j(m)}$)), R(S))), where $y_{i(m-1)} < y \leq y_{j(m)}$; and
(ii) if $P_{j(m-1),j(m+1)}$ is an LR-decreasing (LR-increasing) problem then there exists at least one line segment l in E_h(I) such that l = HLS(y, x(rm($P_{i(m),j(m+1)}$)), R(S)) (HLS(y, L(S), x(lm($P_{i(m),j(m+1)}$)))), where $y_{i(m)} \leq y < y_{j(m+1)}$.
Proof: For brevity the proof is omitted. The proof appears in [6].

□

When all the subproblems are LR-decreasing or LR-increasing, the previous lemma would suffice for our transformation process, because it would associate each segment $l_{i(m)}$ and $l_{j(m)}$ with a unique line segment from E_h(I) in such a way that the sum of their length is greater than X (this is a fundamental property required by our algorithm). However, in general, there are LR-decreasing problems interleaved with LR-increasing problems. For this case the previous lemma does not guarantee the existence of a distinct line segment that could be associated with each $l_{i(m)}$ and $l_{j(m)}$. That is why we need to identify at least two line segments in some regions. Note that in general not all regions have these two line segments, however the two line segments always exist when there is an LR-decreasing problem interleaved with an LR-increasing problem (or vice versa).

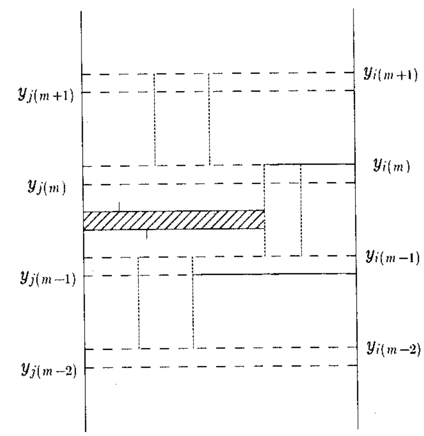


Figure 2.5:

Lemma 6: If $P_{i(m-1),i(m+1)}$ is an LR-decreasing (LR-increasing) problem and $P_{i(m-2),i(m)}$ is an LR-increasing (LR-decreasing) problem, then x(lm($P_{i(m-1),j(m)}$)) = x(lm($P_{j(m-1),j(m)}$)) = x(lm($P_{i(m-1),i(m)}$)) (x(rm($P_{i(m-1),j(m)}$)) = x(rm($P_{j(m-1),j(m)}$)) = x(rm($P_{i(m-1),i(m)}$))) and there are at least two distinct line segments l, l' ∈ E_h(I) such that R(l) = R(l') = x(lm($P_{i(m-1),j(m)}$)) and l, l' ∈ LEFT (L(l) = L(l') = x(rm($P_{i(m-1),j(m)}$)) and l, l' ∈ RIGHT) and $y_{i(m-1)} \leq$ y(l) < y(l') $\leq y_{j(m)}$.
Proof: For brevity the proof is omitted. The proof appears in [6].

□

From lemma 3 and lemma 4 we know that if $l_{i(m)}$ ∈ RIGHT ($l_{i(m)}$ ∈ LEFT) then both $P_{i(m-1),i(m+1)}$ and $P_{j(m-1),j(m+1)}$ are LR-decreasing (LR-increasing) problems. Also from lemma 3 and lemma 4 we know that if $l_{j(m)}$ ∈ LEFT ($l_{j(m)}$ ∈ RIGHT) then both $P_{i(m-1),i(m+1)}$ and $P_{j(m-1),j(m+1)}$ are LR-decreasing (LR-increasing) problems. Let EI = { $l_{i(m)}$ ¦ 1 ≤ m ≤ k } and EJ = { $l_{j(m)}$ ¦ 1 ≤ m ≤ k }. We partition EI into EI_n and EI_c, and EJ into EJ_n and EJ_c as follows:

$$EI_n = \{ l_{i(m)} \mid \text{LEN}(l_{i(m)} \cup l_{j(m)}) < X \};\; EI_c = \{ l_{i(m)} \mid \text{LEN}(l_{i(m)} \cup l_{j(m)}) \geq X \};$$

$EJ_n = \{ l_{j(m)} \mid \text{LEN}(l_{i(m)} \cup l_{j(m)}) < X \}$; and $EJ_c = \{ l_{j(m)} \mid \text{LEN}(l_{i(m)} \cup l_{j(m)}) \geq X \}$.

Obviously, $l_{i(m)} \in EI_n$ ($l_{i(m)} \in EI_c$) iff $l_{j(m)} \in EJ_n$ ($l_{j(m)} \in EJ_c$). If $l_{i(m)} \in EI_c$ then we say the *match for* $l_{i(m)}$ is $l_{j(m)}$. Similarly, if $l_{j(m)} \in EI_c$, the match for $l_{j(m)}$ is $l_{i(m)}$. If l is the match for l' and l' is the match for l, then we say that *l and l' form a matching pair.*

The following procedure finds a match for each of the elements in set $EI_n \cup EJ_n$.

procedure MATCH

ES $\leftarrow \emptyset$;
for m $\leftarrow$ 1 **to** k **do**
  if $l_{i(m)} \in EI_n$ **and** $P_{i(m-1),i(m+1)}$ is a LR-decreasing (LR-increasing) problem **then**
    find a line segment $l \in$ LEFT (RIGHT) such that $l \notin$ ES $\cup$ EI $\cup$ EJ, $y_{i(m-1)} < y(l) \leq y_{j(m)}$,
      and $R(l) \geq x(\text{lm}(P_{i(m-1),j(m)}))$ ($L(l) \leq x(\text{rm}(P_{i(m-1),j(m)}))$);
  Let $l'_{j'(m)}$ = HLS(y(l),L(S),x(lm($P_{i(m-1),j(m)}$))) (HLS(y(l),x(rm($P_{i(m-1),j(m)}$)),R(S)))
  ES $\leftarrow$ ES $\cup$ { $l'_{j'(m)}$ };
  Let $l_{i(m)}$ and $l'_{j'(m)}$ be a matching pair;
endfor

for m $\leftarrow$ 1 **to** k **do**
  if $l_{j(m)} \in EJ_n$ **and** $P_{j(m-1),j(m+1)}$ is an LR-decreasing (LR-increasing) problem **then**
    find a line segment $l \in$ RIGHT (LEFT) such that $l \notin$ ES $\cup$ EI $\cup$ EJ, $y_{i(m)} \leq y(l) < y_{j(m+1)}$,
      and $L(l) \leq x(\text{rm}(P_{i(m),j(m+1)}))$ ($R(l) \geq x(\text{lm}(P_{i(m),j(m+1)}))$);
  Let $l'_{i'(m)}$ = HLS(y(l),x(rm($P_{i(m),j(m+1)}$)),R(S)) (HLS(y(l),L(S),x(lm($P_{i(m),j(m+1)}$))))
  ES $\leftarrow$ ES $\cup$ { $l'_{i'(m)}$ };
  Let $l'_{i'(m)}$ and $l_{j(m)}$ be a matching pair;
endfor

end of procedure MATCH

Lemma 5 and lemma 6 can be used to prove that ES can be constructed, i.e., the line segment with the desired properties can always be found. Let T = EI $\cup$ EJ $\cup$ ES. Let p = $|EI_n|$ (note that $|EJ_n| = |EI_n|$). Since |EI| = |EJ| = k and |ES| = 2 * $|EI_n|$, we know that |T| = 2*k + 2*p. It is simple to verify that | T $\cap$ LEFT| = |T $\cap$ RIGHT| = k + p and T contains k + p matching pairs. Since the length of each matching pair is at least X, we know that LEN(T) $\geq$ (k+p) X.

In phase one of procedure HVH_CUT we introduce a set of horizontal full cuts to partition P into k+p+1 separable subinstances. Remember that set H(I) contains the set of horizontal full cuts introduced in phase one, $H_1(I) = H(I) \cap E_h(I)$, and $H_1'(I) = H(I) - H_1(I)$. We will show that LEN($H_1'(I)$) $\leq$ (k+p) X / 2. In phase three of procedure HVH_CUT we identify a set of guillotine cuts $H_3(I)$. Remember that $H_3(I) \subseteq E_h(I)$. We will prove that LEN($H_1(I) \cup H_3(I)$) $\geq$ (k+p) X. Therefore, LEN($H_1'(I)$) $\leq$ 0.5 * LEN($H_1(I) \cup H_3(I)$) and the 0.5 bound is satisfied.

Let us now order the line segments in T. Note that it is possible that two line segments l and l' of T have the same y-coordinate. However when this happens, the corresponding lines cannot overlap since we are assuming that there are no guillotine cuts. Also, one of the segments is y($l_{i(l)}$) or y($l'_{i'(l)}$) and the other is y($l_{j(l)}$) or y($l'_{j'(l)}$). When we compare two elements with the same y-coordinate value, $l_{j(m)}$ or $l'_{j'(m)}$ is considered smaller than $l_{i(m)}$ or $l'_{j'(m)}$. We sort all line segments in T by their y-coordinates and form the sequence $l_{q(1)}, l_{q(2)}, \ldots, l_{q(2*k+2*p)}$ such that $y(l_{q(1)}) \leq y(l_{q(2)}) \leq \ldots \leq y(l_{q(2*k+2*p)})$. Let $l_{q(0)}$ and $l_{q(2*k+2*p+1)}$ be the bottom and top side of S, respectively.

Lemma 7: $P_{q(w),q(w+2)}$ is separable for $0 \leq w \leq 2*k + 2*p - 1$.
Proof: For brevity the proof is omitted. The proof appears in [6].

□

Let $T_{odd} = \{ l_{q(w)} \mid l_{q(w)} \in T$ and w is odd $\}$ and $T_{even} = \{ l_{q(w)} \mid l_{q(w)} \in T$ and w is even $\}$.

For each line segment $l_{q(w)}$ in T we define its *complement* as the line segment $l^c_{q(w)}$, as follows:

if l(q(w)) $\in$ LEFT (RIGHT) **then**
its *complement* is the line segment $l^c_{q(w)}$ = HLS(y($l_{q(w)}$),R($l_{q(w)}$),R(S)) (HLS(y($l_{q(w)}$),L(S),L($l_{q(w)}$))).

Note that the complement of some line segment in T may overlap with another line segment in T. This can happen only when two line segments in T have the same y-coordinate value. Sets T^c_{odd} and T^c_{even} are defined to have the complements of the elements in sets T_{odd} and T_{even}, respectively.

Given any nonempty non-separable problem P without guillotine cuts, we use the following procedure to partition it.

procedure HVH_CUT

(1) Use MARK and MATCH to construct T;
Order the line segments in T following the rules mentioned above;
Partition T into T_{odd} and T_{even};
if LEN(T^c_{odd}) $\leq$ LEN(T^c_{even}) **then** introduce all the line segments in T^c_{odd}
else introduce all the line segments in T^c_{even};

(2) /* for each resulting partition after step (1), introduce a vertical through cut as follows: */
if LEN(T^c_{odd}) $\leq$ LEN(T^c_{even}) **then** let g = 1 **else** let g = 0;
for w = g **to** 2(k+p) **by** 2 **do**
case
: $l_{q(w+1)}$ is $l_{i(m)}$ or $l_{j(m)}$ for some m :
if $l_{q(w+1)} \in$ LEFT **then** introduce the leftmost vertical through cut in $P_{q(w),q(w+2)}$
else introduce the rightmost vertical through cut in $P_{q(w),q(w+2)}$;
: $l_{q(w+1)}$ is $l'_{i(m)}$ or $l'_{j(m)}$ for some m :
case
/* later on we show the existence of the through cuts we introduce at this step */
: $l_{q(w+1)}$ is $l'_{j(m)}$ for some m **and** $y_{i(m-1)} < y(l_{q(w+1)}) \leq y_{j(m)}$:
if $l_{q(w+1)} \in$ LEFT **then** introduce a vertical through cut with x-coordinate value equal to x(lm($P_{i(m-1),i(m)}$)) in $P_{q(w),q(w+2)}$
else introduce a vertical through cut with x-coordinate value equal to x(rm($P_{i(m-1),i(m)}$)) in $P_{q(w),q(w+2)}$;
: $l_{q(w+1)}$ is $l'_{i(m)}$ for some m **and** $y_{i(m)} \leq y(l_{q(w+1)}) < y_{j(m+1)}$:
if $l_{q(w+1)} \in$ LEFT **then** introduce a vertical through cut with x-coordinate value equal to x(lm($P_{j(m),j(m+1)}$)) in $P_{q(w),q(w+2)}$
else introduce a vertical through cut with x-coordinate value equal to x(rm($P_{j(m),j(m+1)}$)) in $P_{q(w),q(w+2)}$.
endcase
endcase
endfor

(3) **for** each resulting partitions of step (2), introduce a horizontal guillotine cut if possible.

end of procedure HVH_CUT

Let us now establish some important bounds for LEN(H_1(I) $\cup$ H_3(I)) and LEN(H'_1(I)). This bounds are needed to prove Lemma 9. Our result is given by theorem 1.

Lemma 8: Let H_1(I) $\cup$ H_3(I) and H'_1(I) be defined as above. Then,
(i) LEN(H'_1(I)) $\leq$ min{ LEN(T^c_{even}), LEN(T^c_{odd}) } $\leq$ 0.5 * (k+p) * X;
(ii) LEN(H_1(I) $\cup$ H_3(I)) $\geq$ LEN(T_{even}) + LEN(T_{odd}) $\geq$ (k+p) * X; and
(iii) LEN(H'_1(I)) $\leq$ 0.5 * LEN(H_1(I) $\cup$ H_3(I)).

Proof: For brevity the proof is omitted. The proof appears in [6].

□

Lemma 9: $\text{LEN}(E_h'(\text{I})) \leq 0.5 * \text{LEN}(E_h(\text{I}))$ and $\text{LEN}(E_v'(\text{I})) \leq \text{LEN}(E_v(\text{I}))$.
Proof: For brevity the proof is omitted. The proof appears in [6].

□

Theorem 1: $\text{LEN}(E_{ogp}(\text{I})) \leq 1.75\ \text{LEN}(E_{opt}(\text{I}))$.
Proof: For brevity the proof is omitted. The proof appears in [6].

□

DISCUSSION

As pointed out in [2], there is a problem instance I such that $\text{LEN}(E_{ogp}(\text{I})) = 1.5\ \text{LEN}(E_{opt}(\text{I}))$. In this paper we established the bound $\text{LEN}(E_{ogp}(\text{I})) \leq 1.75\ \text{LEN}(E_{opt}(\text{I}))$. We believe that our upper bound cannot be improved by following our proof technique. However, there might be some other way of proving a smaller approximation bound.

REFERENCES

[1] Du, D.Z., and Y.M. Chen, "On Fast Heuristics for Minimum Edge Length Rectangular Partition," Technical Report MSRI 03618-86, Feb. 1986.

[2] Du, D. Z., L. Q. Pan and M. T. Shing, "Minimum Edge Length Guillotine Rectangular Partition," Technical Report, MSRI 02418-86, January 1986.

[3] Gonzalez, T., M.-T. Shing and S.-Q Zheng, "Bounds for Rectangular and Guillotine Partition", Technical Report #TRCS 86-15, CS Dept., UCSB, July 1986.

[4] Gonzalez, T. and S.-Q. Zheng, "Bounds for Partitioning Rectilinear Polygons", Proc. Symp. Computational Geometry, June 1985, pp. 281-287. (also appears as Technical Report #85-22, CS Dept., UCSB, Dec. 1985).

[5] Gonzalez, T. and S.-Q. Zheng, "Approximation Algorithms for Partitioning Rectilinear Polygons", Technical Report # 85-21, CS Dept., UCSB, 1985.

[6] Gonzalez, T. and S.-Q Zheng, "Improved Bounds for Rectangular and Guillotine Partition", Technical Report #TRCS 86-16, CS Dept., UCSB, July 1986.

[7] Lingas, A., "Heuristics for Minimum Edge Length Rectangular Partitions of Rectilinear Figures", 6th GI-Conference, Dortmund, 1983, Lecture Notes in Computer Science, 195 (Spring-Verlag).

[8] Lingas, A., R. Y. Pinter, R. L. Rivest and A. Shamir, "Minimum Edge Length Partitioning of Rectilinear Polygons", Proc. 20th Annual Allerton Conference on Communication, Control and Computing, Monticello, Illinois, Oct. 1982.

[9] Levcopoulos, C., "Minimum Length and Thickest-First Rectangular Partitions of Polygons," Proceedings of the 23rd Allerton Conference on Communication, Control and Computing, U. of Illinois, Oct. 1985.

[10] Levcopoulos, C., "Fast Heuristics for Minimum Length Rectangular Partitions of Polygons," Proceedings of the 2nd Computational Geometry Conference, June 1986.

[11] Rivest, R. L. "The 'PI' (Placement and Interconnect) System", Proc. 19th Design Automation Conference, June 1982.

[12] Shing, M.T., private communication.