On The Generalized Channel Definition Problem

Teofilo Gonzalez

Department of Computer Science
The University of California
Santa Barbara, CA 93106

Mohammadreza Razzazi

Computer Engineering and Science
Case Western Reserve University
Cleveland, Ohio 44106

Abstract

The generalized channel definition problem has been modeled as the following partition problem. Let RP be a boundary defined by a rectilinear polygon in E^2 and let H be a set of holes defined by disjoint rectilinear polygons inside RP. For $IP = (RP, H)$, we use $p(IP)$ to denote the length of the line segments that define RP plus the sum of the length of the line segments that define the holes in H. We consider the $RP - RP$ problem in which RP is partitioned into rectangles by introducing a set of orthogonal line segments with least total length. We use $m(IP)$ to denote the total length of the partitioning segments in an optimal solution to IP. The problem of finding $m(IP)$ given IP is NP-hard. In this paper we present an $O(n \log n)$ approximation algorithm for the $RP - RP$ problem that generates solutions with length at most $2.5p(IP) + 6m(IP)$, where n is the total number of segments in RP and H.

1 Introduction

The input to the generalized channel definition problem is a rectilinear polygon RP in 2-dimensional Euclidean space (E^2), and a set of disjoint rectilinear polygons H defined inside RP. The set of rectilinear polygons H represents modules with terminal points defined on their perimeter that need to be interconnected. The area of the rectilinear polygon RP that is not occupied by the set H represents the routing area. To solve the routing problem we partition the routing area into rectangles (channel definition), specify the channels or rectangles that each net crosses (global routing) and then route in each of these channels or rectangles (detail routing). In this paper we study the *generalized channel definition problem* ($RP - RP$), defined as partitioning the routing area ($RP - H$) into rectangles such that the total length of the partitioning line segments is least possible. This measure minimizes the total perimeter of the rectangle routing problems to be solved, and as a result of this, it tends to reduce their complexity [12].

For problem instance $IP = (RP, H)$, we use $p(IP)$ to denote the total length of the line segments that define RP plus the sum of the length of the line segments that define the holes in H. We use $m(IP)$ to denote the total length of the partitioning segments in an optimal solution to IP. The problem of finding $m(IP)$ given IP is NP-hard. In this paper we present an $O(n \log n)$ approximation algorithm for the $RP - RP$ problem that generates solutions whose line segments have total length bounded by $2.5p(IP) + 6m(IP)$, where n is the total number of line segments in RP and H.

Several approximation algorithms for the $RP - RP$ have been developed. In table I we list all known approximation algorithms for this problem. The best of these algorithms are the ones reported in [10] and [11]. Levcopoulos conjectured that the bound of $8m(IP)$ in [10] can be reduced to $4.7m(IP)$, the bound $cte\ m(IP)$ in [11] can be reduced to $5m(IP)$ and the bound $12m(IP) + 6p(IP)$ in [11] can be reduced to $7m(IP) + 3.5p(IP)$.

Table I: Approximation Algorithms for the $RP - RP$ Problem

Reference	Approximation	Time
[12], [13]	-	fast
[8]	$41m(IP)$	$O(n^4)$
	$(4m(IP) + p(IP))/2$	$O(n^4)$
[1]	$37m(IP)$	$O(n^2 \log n)$
	$21m(IP)$	$O(n^3)$
	$11m(IP)$	$O(n^4)$
[10]	$8m(IP)$	$O(n^2)$
[11]	$cte\ m(IP)$	$O(n \log n)$
	$12m(IP) + 6p(IP)$	$O(n \log n)$

Levcopoulos' algorithm [11] consist of two parts. In the first part a rectangle R (whose sides are orthogonal to the x and y axes) is placed around RP, and each of the concave corners of RP and the convex corners of H is replaced by a point. The set of all such points is referred to by P. That is, from an instance of the $RP - RP$ problem IP we construct an instance $I = (R, P)$ of the $RG - P$ problem. The objective of the $RG - P$ problem is to introduce a set of line segments of least total length that partitions R into rectangles that do not contain points from P inside them. This problem is solved by an approximation algorithm similar to that in [5] (other algorithms for the $RG - P$ problem are given in [3], [5], [6], and [4]). The approximation bound generated for IP in this case is $12m(IP) + 6p(IP)$, but as mentioned above, Levcopoulos conjectures that it is bounded by $7m(IP) + 3.5p(IP)$. Obviously, this approach is advantageous when $p(IP) < m(IP)$. The second part deals with the case when $p(IP) \geq m(IP)$. In this case a thickest first heuristic similar to the one developed in [10] is used to generate a near-optimal solution.

In this paper we develop for the $RP - RP$ problem an approximation algorithm that generates solutions whose segments have total length bounded by $6m(IP) + 2.5p(IP)$. Since the algorithm in [11] depends on this case, our new approximation bound may be used to reduce the worst case approximation bound for the algorithm in [11]. Our algorithm is robust in the sense that it can be generalized to solve a more general version of the problem, i.e., when it is defined over E^d for $d > 2$. For this case the time complexity bound is $O(dn\ log\ n)$ and the solutions generate have total $d-1$ volume at most $(2d-1.5)p(IP) + (4d-2)m(IP)$.

2 Approximation Algorithm

Let us now discuss our approximation algorithm for the $RP - RP$ problem. First a rectangle R (whose sides are orthogonal to the axes) is placed to include the rectilinear polygon, and each of the concave corners of RP and convex corners of H is replaced by a point. The set of points is referred to as set P. In other words, from an instance $IP = (RP, H)$ of the $RP-RP$ problem we construct an instance $I = (R, P)$ of the $RG-P$ problem. This problem is solved by the divide-and-conquer procedure given in [3]. All the parts of the line segments introduced by that algorithm inside RP but not inside the set of holes H are said to form the solution to the original problem, if it is the case that the segment includes a point in P or there is another segment introduced by the divide-and-conquer algorithm incident to it. We should point out that if we do not place these two restrictions, then there could be about n^2 segments in our solution. By placing these two restrictions we limit the number of segments to $O(n)$. The resulting segments are referred to as $E_{apx}(IP)$ and the segments introduced by the divide-and-conquer procedure that are inside RP but outside H are referred to as $SET(IP)$. Given a set of line segments S, we use $V(S)$ to denote the sum of the length of the lines segments in S. It is simple to see that $V(SET(IP)) \geq V(E_{apx}(IP))$.

Before we analyze the performance of our algorithm in more detail, we formally define the $RG-P$ problem. An instance of the $RG-P$ problem is given by $I = (R = (O, X), P)$, where O and X define a rectangle or boundary R ($O = (o_1, o_2)$ is the "lower-left" corner of the rectangle (*origin* of I), and $X = (x_1, x_2)$ are the dimensions of the boundary) in 2-dimensional Euclidean space (E^2), and $P = \{p_1, p_2, ..., p_n\}$ is a set of points (degenerate holes) inside rectangle R. The objective is to introduce a set of line segments of least total length such that each point in P is in at least one of the partioning line segments. We shall refer to the two dimensions (or axes) of E^2 by the integers 1 and 2 (the first dimension (x-dimension, x-axis, or 1-axis) and the second dimension (y-dimension, y-axis, or 2-axis)).

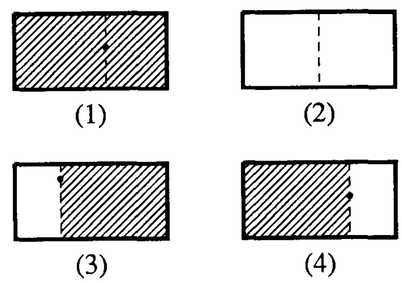

Figure 1: Subproblems generated by PARTITION.

Let us now explain procedure PARTITION, the divide-and-conquer approximation algorithm for the $RG-P$ problem given in [3]. Procedure PARTITION begins by rotating the rectangle so that $x_1 \geq x_2$. Then it checks if $P(I)$ is empty and if so, it returns. Otherwise, it introduces a *mid-cut* or an *end-cut*. A *mid-cut* is a line segment orthogonal to the 1-axis that intersects the center of the rectangle (i.e., it includes point $(o_1 + x_1/2, o_2 + x_2/2)$) and an *end-cut* is a line segment orthogonal to the 1-axis that contains either the "leftmost" or the "rightmost" point in $P(I)$. A *mid-cut* is introduced when the two resulting subproblems have at least one point each. Otherwise, an *end-cut* is introduced. The *end-cut* intersects the leftmost point if such a point is not located to the left of the center of the rectangle, otherwise the *end-cut* intersects the rightmost point.

It is easy to verify that figure 1 represents all the possible outcomes of one step in the recursive process of our algorithm. A shaded region in figure 1 represents a subinstance without interior points.

3 Analysis

Now let us analyze the performance of our algorithm. To establish the time complexity bound is simple since it follows from the fact that the number of points is n; the time complexity bound in [3]; and the fact that $O(n)$ segments are introduced. Now let us concentrate on the approximation bound. Let $E_{opt}(IP)$ be any optimal rectangular partition for IP. Let R' be any rectangle in it. Note that all the area of R' must also be inside RP. When running the algorithm we may think of the problem instance as being formed by $I' = (R, R', P)$, even though the algorithm does not know R'. Let $B(R'(I))$ be the length of the sides in $R'(I)$ located inside $R(I)$, and let $OV(R'(I))$ be length of the sides of $R'(I)$ inside $R(I)$ that overlap with the line segments introduced by PARTITION. In section 4 we establish that the set of segments introduced by the algorithm inside or on R', which we denote by $SET(R'(I))$ is such that

$$V(SET(R'(I))) \leq 2.5B(R'(I)) + OV(R'(I)).$$

Summing over all R',

$$\sum V(SET(R'(I))) \leq \sum 2.5B(R'(I)) + OV(R'(I)).$$

Since $\sum B(R'(I)) = 2m(IP) + p(IP)$, and $\sum OV(R'(I))$ is $m(IP)$ (note that the only new segments that overlap with the boundary are those in $E_{opt}(IP)$), we know that

$$V(E_{apx}(IP)) \leq 6m(IP) + 2.5p(IP).$$

These facts are summarized by the following theorem.

Theorem 1: $V(E_{apx}(IP)) \leq 6m(IP) + 2.5p(IP)$.

Proof: By the above discussion.

4 Bound for $V(SET(R'(I)))$

Let us now show that for every problem instance I defined above, algorithm PARTITION introduces a set of line segments inside $R'(I)$ whose total length is at most $2.5B(R'(I)) + 3.5OV(R'(I))$. In lemma 1 we prove a stronger result (simpler to prove) that uses the *CARRY* function defined below. One may visualize our proof as follows. Every time a line segment is introduced inside R' by the algorithm it is colored red. The segments in R' that overlap with a cut (OV) or segments in B belong to an instance without points, are colored blue. Our approach is to bound the length of the red segments by that of the blue segments. The segments in *SET* represent the segments introduced in R' and *CARRY* represents some previously introduced red segments that have not yet been accounted for by blue segments. The proof consists of showing that at all times the length of the red segments can be accounted by that of the blue segments. Let us develop some precise technical notation required for the definition of the *CARRY* function.

A problem instance I consists of an exterior rectangle defined by $R = (O, X = (x_1, x_2))$, where O is a vector defining the origin point and X is a vector of dimensions; an interior rectangle defined by $R' = (O', X' = (x_1', x_2'))$, where O' is a vector defining the origin point and X' is a vector of dimensions; and a set of interior points $P(I)$. When procedure PARTITION introduces a cut c (*mid-cut* or an *end-cut*) it partitions the problem I into two subproblems denoted by I_a and I_b. Each of these subproblems consists of an exterior rectangle, an interior rectangle and a set of interior points. When referring to problem instance K, where $K \in \{I, I_a, I_b\}$, we use $R(K) = (O(K), X(K) = (x_1(K), x_2(K)))$ to refer to the exterior rectangle; $R'(K) = (O'(K), X'(K) = (x_1'(K), x_2'(K)))$ to refer to the interior rectangle; and $P(K)$ to refer to the set of interior points in K. We define $SET(R'(I))$ as the total length of the line segments introduced inside $R'(K)$. In what follows we refer to the two dimensions of E^2 by the integers 1, and 2. For $K \in \{I, I_a, I_b\}$ and $1 \leq j \leq 2$, let $F_j(R(K))$ ($F_j(R'(K))$) be the set of sides in $R(K)$ ($R'(K)$) orthogonal to the j-axis. For $K \in \{I, I_a, I_b\}$ and $1 \leq j \leq 2$, let

$$f_1(R'(K)) = x_2'(K) \text{ and } f_2(R'(K)) = x_1'(K),$$

be the length of the sides in $F_j(R'(K))$; and let

$$f_1(R(K)) = x_2(K) \text{ and } f_2(R(K)) = x_1(K)$$

be the length of each of the sides in $F_j(R(K))$. The sum of the length of the sides of $R'(K)$ located inside $R(K)$ (not on the boundary of $R(K)$) is denoted by $B(R'(K))$; the sum of the length of the sides of $R'(K)$ that are inside $R(K)$ (not on the boundary of $R(K)$) and overlap with all cuts introduced by the algorithm is denoted by $OV(R'(K))$; the sides of $R'(K)$ that overlap with the sides of R are called *exposed sides* of $R'(K)$, otherwise they are called *non-exposed sides.* Suppose that PARTITION introduces a *mid-cut* or *end-cut* c in I that partitions the problem into subproblems I_a and I_b, then the length of the segments introduced inside $R'(I)$ is denoted by $USE(I, c)$ and the length of the cut that overlaps with a boundary of $R'(I)$ is denoted by $OVC(I, c)$. Clearly,

$$B(R'(I)) = B(R'(I_a)) + B(R'(I_b)) + OVC(I, c), \text{ and}$$

$$OV(R'(I)) = OV(R'(I_a)) + OV(R'(I_b)) + OVC(I, c).$$

Problem instance $K \in \{I, I_a, I_b\}$ is said to be of type E if $R'(K) = \emptyset$, and it is said to be of type $L(K) = (l_1(K), l_2(K))$ if $l_j(K)$ sides in $F_j(R'(K))$ are *exposed*, for all $1 \leq j \leq 2$. We say that a problem instance is of type $L + 1$ if it is of type $(l_1 + 1, l_2)$ or $(l_1, l_2 + 1)$. Note that for some Ls, $L + 1$ does not have any interpretation.

We say that an *exposed* side in $F_j(R'(K))$ is a *regular side* if either of the following two conditions hold:

- there is a $non-exposed$ side in set $F_j(R'(K))$; or
- $f_j(R'(K)) \leq 2f_p(R'(K))$ for all $1 \leq p \leq 2$ such that there is a $non-exposed$ side in $F_p(R'(K))$, and there is at least one $non-exposed$ side in $R'(K)$.

Otherwise, we say that it is an *irregular* side. Note that sides are *exposed* or $non-exposed$. An *exposed* side can be a *regular* or an *irregular* side. A *regular* or *irregular* side must be *exposed*. Note that if all sides are *exposed* then all sides are *irregular*. It is not important to identify the type of sides when $R^{'}(K)$ is $\emptyset$.

If $R^{'}(K)$ is not $\emptyset$ and there is at least one $non-exposed$ side in $R^{'}(K)$, we define $mn(K)$ as w such that $F_w(R^{'}(K))$ contains a $non-exposed$ side of least length amongst all $non-exposed$ sides in $R^{'}(K)$; otherwise $mn(K)$ is undefined. Note that if there is an *exposed* side in $F_j(R^{'}(K))$ then all the *exposed* sides in it are either *regular* or *irregular*. If $R^{'}(K)$ is not $\emptyset$ and there is at least one $non-exposed$ side in $R^{'}(K)$, we define $CARRY(K,j)$ as $2f_{mn}(K)(R^{'}(K))$ if the sides in $F_j(R^{'}(K))$ are *irregular*; as $0.5l_j(K)$ $f_j(R^{'}(K))$ if there is a *regular* side in $F_j(R^{'}(K))$. Otherwise it is defined as zero. We define $CARRY(K)$ $= \sum_{j=1}^{2} CARRY(K,j)$.

Let I be a problem instance that has been partitioned by a *mid-cut* or an *end-cut* c orthogonal to the 1-axis into subproblems I_a and I_b. If the cut c overlaps with a side of $R^{'}(I)$, then either $R^{'}(I_a) = R^{'}(I)$ or $R^{'}(I_b)$ is $\emptyset$, or vice-versa. Before proving our main result, a couple of technical lemmas need to be established. For brevity we do not include them. The main result of the section is given in lemma 1.

Lemma 1: For any problem instance I,

$$SET(R^{'}(I))+CARRY(I) \leq 2.5B(R^{'}(I))+OV(R^{'}(I)).$$

Proof: The proof is by induction on the number of points inside I, i.e., $|P(I)|$. For brevity the proof is omitted. An interested reader can find the proof in [2]

5 Summary and Conclusions

We have developed an $O(n \log n)$ approximation algorithm for the $RP-RP$ problem that generates solutions whose line segments have total length at most $2.5p(IP)+6m(IP)$. Obviously, when $p(IP) \leq m(IP)$, it generates reasonably good solutions. Our approximation bound degrades when $p(IP)$ is very large compared to $m(IP)$.

We have also considered a more general version of the problem, i.e., when it is defined over E^d for $d > 2$. Our algorithm can be easily generalized for this more general case [2]. The time complexity for the algorithm becomes $O(dn \log n)$ and it generates solutions whose $d-1$ volume is at most $(2d-1.5)p(IP)+(4d-2)m(IP)$. The proof has a couple of additional cases more than the one for the case when $d = 2$. Obviously, when $p(IP) \leq m(IP)$, our algorithm generates reasonably good solutions. Our approximation bound degrades when $p(IP)$ is very large compared to $m(IP)$. Developing an efficient approximation algorithm for this other case seems difficult. The approach used for the case when $d = 2$, cannot be generalized to arbitrary d.

References

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

[2] T. Gonzalez, and M. Razzazi, "An Approximation Algorithm for Partitioning a Hyperrectilinear Polytope with Holes," UCSB Technical Report TRCS 90-23, November, 1990.

[3] T. Gonzalez, M. Razzazi, and S. Zheng, "An Efficient Divide-and-Conquer Approximation Algorithm for Hyperrectangular Partitions," Second Canadian Computational Geometry Conference, August 1990, 214 - 217, (an extended version appears in UCSB Technical Report TR-89-24).

[4] T. Gonzalez, M. Razzazi, M. T. Shing, and S. Zheng, "On Optimal Guillotine Partitions Approximating Hyperrectangular Partitions," UCSB Technical Report TRCS 89-25, October 1989.

[5] T. Gonzalez, and S. Q. Zheng, "Bounds for Partitioning Rectilinear Polygons", Proceedings of the Symposium on Computational Geometry, June 1985, pp. 281-287, (an extended version appears in UCSB Technical Report TR-85-22).

[6] T. Gonzalez, and S. Q. Zheng, "Approximation Algorithms for Partitioning Rectilinear Polygons" , *Algorithmica*, 5, January 1990, 11 - 42.

[7] T. Gonzalez, and S. Q. Zheng, "Improved Bounds for Rectangular and Guillotine Partitions," *Journal of Symbolic Computation*, 7, 1989, 591 - 610.

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

[9] A. Lingas, 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.

[10] C. Levcopoulos, "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.

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

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

[13] J. D. Ullman, *Computational Aspects of VLSI*, Computer Science Press, 1984.