Minimum Edge Length Partitioning of Rectilinear Polygons†

(Extended Abstract)

Andrzej Lingas*
Ron Y. Pinter*
Ronald L. Rivest*
Adi Shamir**

Abstract

We consider the problem of partitioning a rectilinear polygon into disjoint rectangles using the minimum amount of "ink". The given polygonal figure may or may not have holes in it, and the complexity of the problem is shown to depend on this very fact: For hole-free figures, we provide a polynomial-time algorithm, whereas for figures containing holes the problem is shown to be NP-complete.

For the hole-free case, the time complexity of the partitioning algorithm is $O(n^4)$ for arbitrary rectilinear shapes, but for non-trivial special cases the complexity can be reduced to $O(n^3)$. In the general case it turns out that even degenerate holes, *i.e.* points through which the decomposing edges have to go, are enough to render the problem intractable.

More interestingly, the proof techniques for the NP-completeness results can be applied to an even more general case, namely where the polygons are no longer rectilinear and a partitioning into convex polygons with minimum edge length is sought.

1. Introduction

The problem of dividing a polygonal region into parts has received considerable attention recently ([ChDo79], [LLMP79], [LLMPL79], [GJPT78]). Traditionally, convexivity was the overriding issue and the goal was to obtain a convex partition using the minimum number of components. In this paper we focus on *three* new aspects of the problem, two regarding shape, and one — the minimality criterion: First we consider only *rectilinear* polygons, *i.e.* figures with "vertical" and "horizontal" boundary edges; such polygons can always be *partitioned* into rectangles (with the same orientation as the original figure). Second we allow the inclusion of *holes* (or *islands*) in the given figure, *i.e.* one enclosing polygon may contain several polygonal shapes cut out of it (see Figure 1). Finally, we introduce and investigate a new minimality criterion, namely — the total length of the *edges* (lines) used to form the internal partition; we call this criterion *minimization of edge length.*

† This research was supported by DARPA grant N00014-80-C0622, NSF grant MCS-8006938, and an IBM graduate fellowship.

* Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Mass. 02139

** Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel.

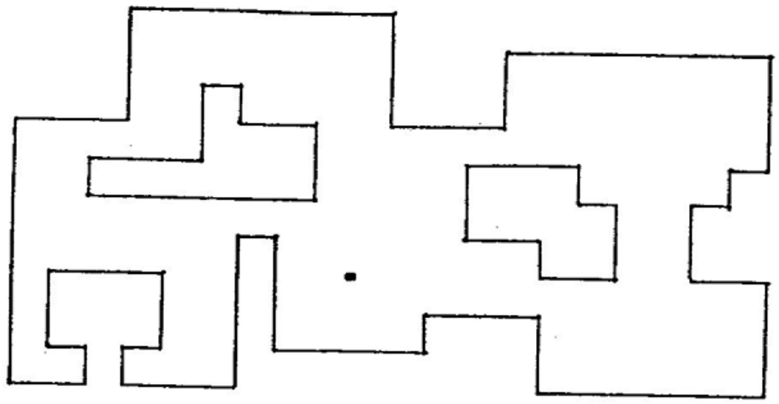
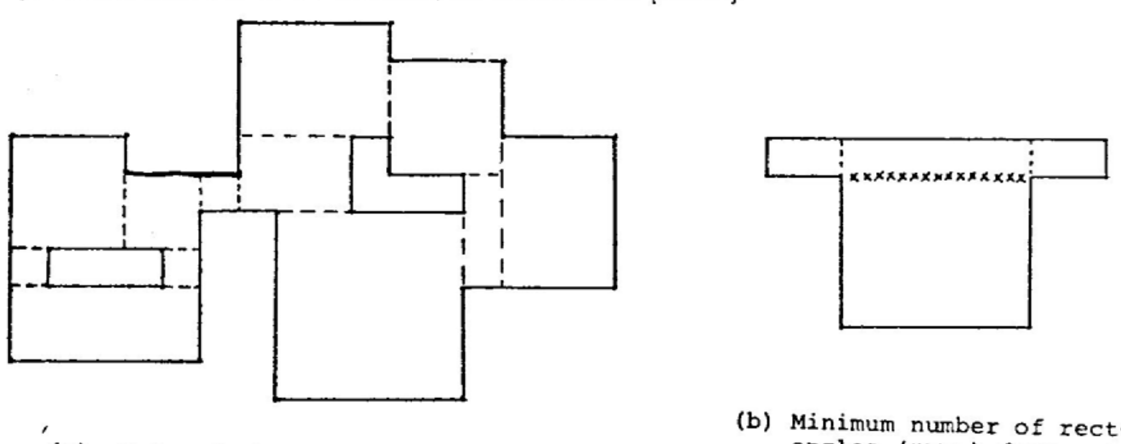


Figure 1. A rectilinear figure with three holes (one is degenerate).

We investigated various versions of the problem of minimizing the edge length partitioning of a rectilinear figure. The major result of this paper is that the critical characteristic of an instance of the problem is whether it has holes or not: If it does not, a polynomial time algorithm exists to solve it optimally; otherwise, the problem is NP-complete ([GaJh79]) even if the holes are degenerate, *i.e.* they are points through which the partitioning edges have to pass. Interesting polynomial-time heuristics that can achieve solutions whose edge length is within a constant factor from optimal are discussed in [Lin81].

The problem of finding optimal partitions of such figures with respect to our criterion has applications in several domains: Process control (stock cutting), automatic layout systems for integrated circuits (channel definition), and even 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 ([Ri82]) 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. At an intuitive level, these partitions might be considered analogous to the way rectilinear soap-bubbles would decompose a given space into convex parts.

Notice that the traditional minimality criterion of reducing the *number* of components — which leads to different applications — does not coincide with ours. Figure 2(b) shows a trivial (and generic) example of this phenomenon. Also, the relative difficulties of the various classes of problems turn out to be different, as discussed in [Lin82].



(a) Holes interact with boundary

(b) Minimum number of rectangles (xxxx) does not coincide with minimum edge length (---).

Figure 2. Minimum edge length decompositions into rectangles.

In what follows, we first set up some notation (Section 2). Then we present the polynomial-time algorithm for the hole-free cases (Section 3) and sketches of the NP-completeness proofs for the general case (Section 4). Finally, we pose extensions and open problems (Section 5).

2. Notation and Definitions

A rectilinear *boundary* is a simple polygon, all of whose sides are either parallel or perpendicular to each other. More precisely, each side is a segment of a line in the plane, and the set of lines corresponding to a boundary can be partitioned into two sets such that within each set all lines are parallel to each other, and the two sets are mutually perpendicular. We arbitrarily call one direction *horizontal*, the other *vertical*. Clearly, all *corners* of a boundary are either of 90^0 (*convex*) or 270^0 (*concave*). A boundary may be represented by a list of its corners as points in the plane, $P_1, P_2, \ldots, P_n$, in (say) clockwise order. (Simplicity can then be checked in $O(n \log n)$ time ([ShHo76])).

A rectilinear *hole* in a rectilinear boundary is a simple rectilinear polygon fully enclosed by it. We allow only holes whose orientation is the *same* as that of the enclosing boundary, *i.e.* the horizontal and vertical directions corresponding to the polygons coincide. A hole is represented as before by a list of points describing its circumference; a hole is called *degenerate* if it consists of a single point.

A rectilinear *figure* is a set of rectilinear boundaries, each of which may contain an arbitrary number of non-overlapping holes in it. Notice that holes are not allowed to contain further holes. From here on, we shall consider (without loss of generality) only figures consisting of one boundary and its holes; for the problems considered in this paper, an optimal solution for a set of boundaries may be obtained by solving the problem for each member separately. Figure 1 shows a typical figure consisting of a boundary with several holes in it.

A *rectangular partition* of a figure is a set of line segments, $l_1, l_2, \ldots, l_m$, lying within its boundary and not crossing any holes, so that when drawn into the figure the area not enclosed by holes is partitioned into rectangles; clearly, no "loose ends" are allowed. Figure 2(a) is an example of such a partition. We could have given a formal definition of what it means for a set of line segments to partition an area (with holes) into rectangles, but it would not serve any useful purpose in the extext of this abstract.

Finally, the partitioning line segments are called *edges*. A minimum *edge length* partitioning of a figure is a rectangular partitioning such that the sum of lengths of its edges is smallest among all valid partitionings. Incidentally, this is equivalent to requiring that the total of all perimeters of the resulting rectangles be minimized.

3. A Polynomial-time Partitioning Algorithm for Hole-free Figures

In this section we show that a minimum edge length partitioning of a rectilinear figure consisting only of a simple boundary (henceforth *hole-free*) can be found in polynomial time. The complexity in the general hole-free case is $O(n^4)$ (where n is the number of sides on the boundary), but in case the boundary has a shape of a *histogram* (to be defined in due course), an $O(n^3)$ algorithm exists.

The algorithm used makes use of dynamic programming (see, for example, [AHU74]) — a rather natural approach for this problem: minimum edge length triangulation can be easily solved using the this technique. The difficult part of the proof for the rectangular case involves showing how a given polygon can be split in a small number of ways while guaranteeing that the optimal partitioning is consistent with one of these splits.

Before we proceed, let us add two definitions used only in this section: First, the *grid* induced by a boundary is the set of horizontal and vertical lines (and their intersection points) on which segments of the boundary lie. Second, a *constructed line* is a maximal extension of a partitioning edge to include any sides of the boundary that are contiguous to the edge and go in the same direction. Two or more aligned edges may be put on the same constructed line, but this is not always the case. This is exemplified in Figure 3 where constructed lines are dotted, whereas edges are dashed.

The following two lemmas serve to center our attention on the crux of the matter:

Lemma 1. In all minimum edge length solutions for any given boundary $\mathcal{B}$, all edges and internal corners lie on the induced grid.

Lemma 2. For any given boundary $\mathcal{B}$, there exists a minimum edge length partitioning such that every constructed line has at least one end on the original boundary $\mathcal{B}$.

Ends of constructed lines that lie on the boundary are called *anchors;* thus Lemma 2 can be restated as saying that every constructed line has at least one anchor. An anchor may be a corner or a point on a side perpendicular to the edge. In Figure 3 we marked anchors with solid circles; in 3(a) an optimal solution satisfying the lemma is shown, whereas in 3(b) we see that there exist figures for which all constructed lines can have at most one anchor in order to obtain optimality. Thus Lemma 2 cannot be strengthened.

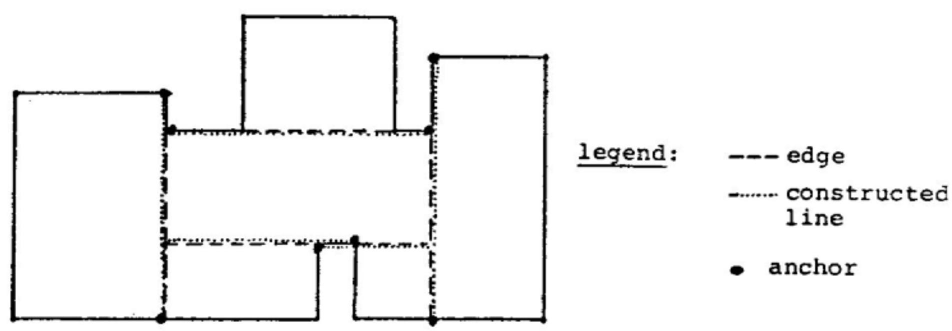


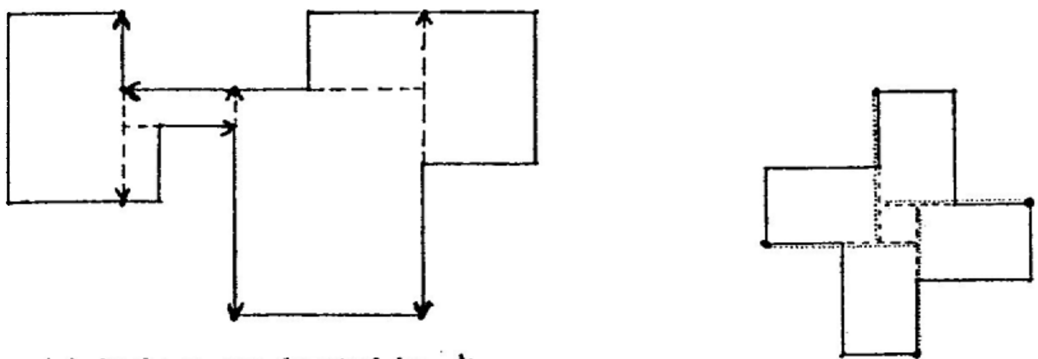
Figure 3. The constructed lines of a decomposition.

In the process of finding the optimal solution, we shall look at portions of the original figure which are in themselves hole-free boundaries. We claim that it is enough to look at portions whose boundary consists of a contiguous piece of the original boundary and *at most* two constructed lines, that again, are contiguous. In other words, the most complicated intermediate figure looks like a piece of the original boundary cut off by a (convex or concave) L-shaped pair of constructed edges; sometimes it is just cut off by one edge, and initially it is not cut off at all. Figure 5 shows two such intermediate figures. We shall show that our algorithm generates only intermediate figures satisfying the above property, and this in fact is the invariant preserved by all partitioning rules.

Now we have to devise the rule for partitioning: For each intermediate figure we define a *candidate* point depending on the number of constructed lines it has: If 0, then any convex corner may be chosen; if 1, then either end of the constructed line is chosen; and if 2, then we have to choose the common end of the two constructed lines (remember they are adjacent).

The *matching* point is defined as one lying on the induced grid, and such that there exists a (not necessarily optimal!) partitioning of the partial figure in which the candidate point and the matching point are kitty corners of a rectangle. Figure 5 shows forbidden cases of matching points. Furthermore, in order to maintain optimality, the matching point is restricted to lie in certain areas with respect to the candidate point as characterized by Lemmas 3 and 4 below.

Note that without loss of generality, we can always assume that the partial figure has two



(a) Anchors are denoted by

(b) A "pinwheel": only one anchor per const. line.

Figure 4. There exist optimal solutions in which every constructed line is anchored.

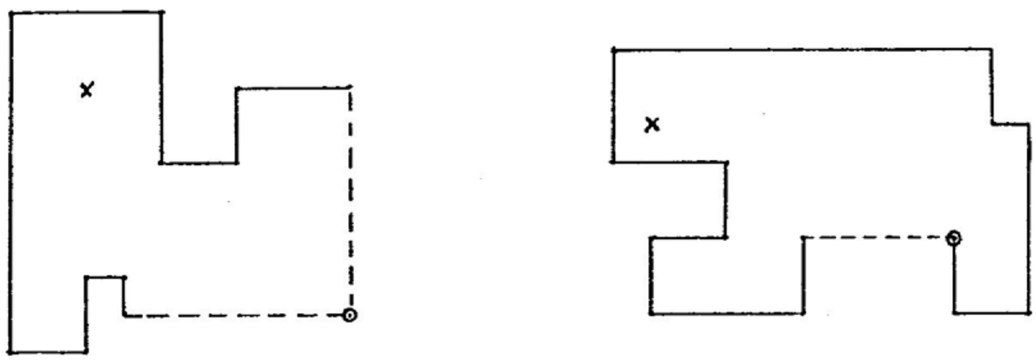


Figure 5. Forbidden matching points. ⊙ = candidate point
X = matching point

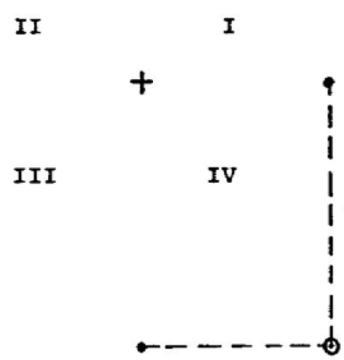


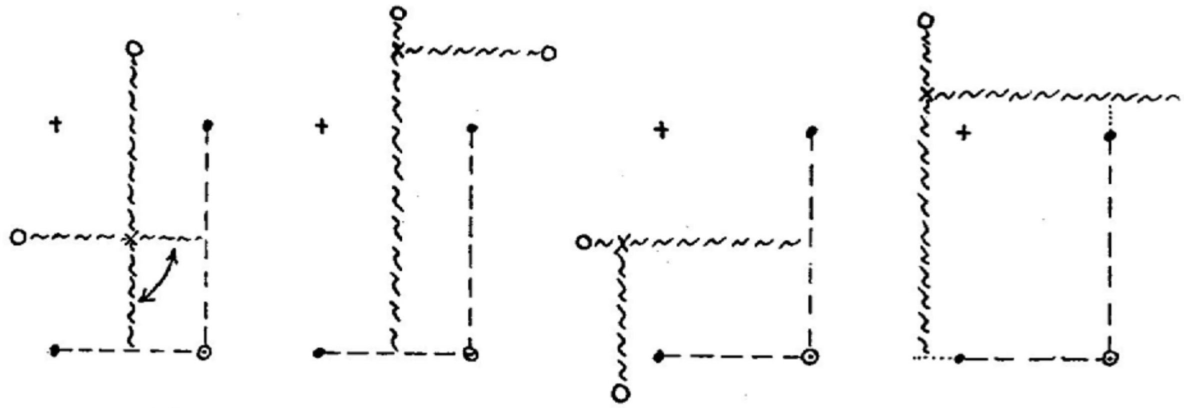
Figure 6. The four quadrants around the kitty-corner origin (+).

constructed lines and that the candidate point is at their common end. Now the candidate point can be either convex or concave with respect to the rest of the partial figure.

First let us assume it is convex. Then let us define the *kitty-corner origin* as the kitty corner of the rectangle that would have been formed by using the two constructed lines as its sides (see Figure 6). There are four quadrants defined with respect to this origin. We claim that

Lemma 3. The matching point of a convex candidate cannot reside in the quadrant including the candidate point (IV in the Figure 6). Whenever it resides in one of the other quadrants, the basic invariant is preserved.

The first part is proven by realizing that one edge will be superfluous in any solution placing the matching point in this quadrant (Figure 7(a)), thus defying its optimality. The second part is proven by inspection, showing that the only possible partitions that will not violate minimality



(a) (b) (c) (d)

Figure 7. The proof of Lemma 3.

legend:
--- existing constructed line
• their anchors
⊙ candidate point
+ kitty-corner origin
x matching point
~~~ new constructed lines
o their anchors
... extensions to existing constructed lines

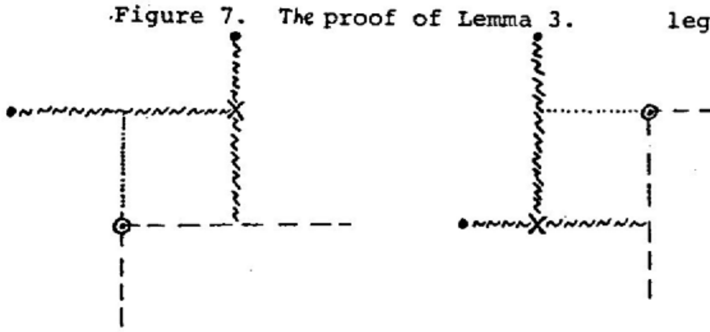


Figure 8. Proof of Lemma 4; same legend as in Figure 7.

are as shown schematically in Figure 7(b),(c) and (d). Notice that constructed lines are extended in the process in a manner consistent with their definition.

If the candidate point is concave, the situation is simpler:

Lemma 4. The matching point of a concave candidate must have a horizontal or vertical projection on one of the constructing edges.

The proof is similar to that of Lemma 3, and is illustrated in Figure 8.

For each candidate point there are $O(n^2)$ possible matching points, and the (intermediate) minimum edge length partitioning resulting from each match has to be evaluated. But since the induced grid has $O(n^2)$ points, there are only that many intermediate figures looked at during the whole algorithm. If we maintain a table of all such possible figures (sorted, say, by area), we can apply dynamic programming successfully to obtain a global minimum.

Putting everything together, we obtain

Theorem 1. Minimum edge length partitioning of a hole-free figure can be found in time $O(n^4)$.

If the hole-less figure can be described by a list $P_1, P_2, \ldots, P_n$ such that the x-coordinates of the points form a monotonically non-decreasing sequence, or — alternatively — the y-coordinates have this property, then we say that the figure is a *histogram*. In this case, a faster algorithm exists, which is easiest to visualize when histograms are being drawn upside down, as in Figure 9; the long edge at the top is the *sky* and all sides parallel to it are *roof tops*.

We take the roof top closest to the sky, and at each of its corners (unless it is at the left or right end) we have to decide whether to draw a vertical or a horizontal edge. We can show that one of these edges (in its entirety) appears in a minimum edge length solution. Again, we use dynamic programming to keep track of optimality of partial figures. For convergence, notice that the sky is lowered for a contiguous portion of the histogram each time a horizontal edge is drawn, or we are left with a shorter histogram if a vertical edge is drawn. This procedure is faster than the general one, and we have
~~~

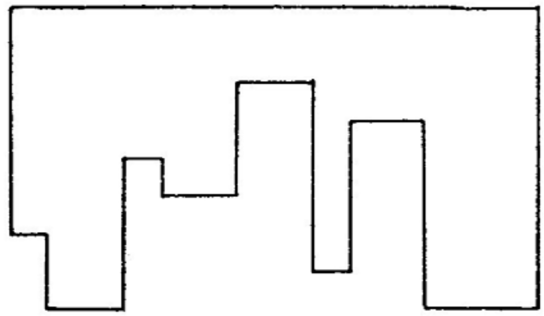


Figure 9. An upside-down histogram.

Theorem 2. Minimum edge length partitioning of a histogram can be found in time $O(n^3)$.

4. Partitioning Figures with Holes is NP-complete

In this section we show that the following decision problems are strongly NP-complete:

(1) **Minimum Edge Length Rectangular Partitioning (MELRP):** Given a rectilinear figure as defined in Section 2 and a number k, is there a rectangular partitioning whose edge length does not exceed k?

(2) **Minimum Edge Length Rectangular Partitioning with Points (MELRPP):** Given a rectilinear figure as defined in Section 2 all of whose holes are degenerate, and a number k, is there a rectangular partitioning whose edge length does not exceed k?

(3) **Minimum Edge Length Convex Partitioning (MELCP):** Given a simple polygon with non-overlapping, polygonal holes in it, and a number k, is there a partitioning of its area into convex parts whose edge length does not exceed k?

(4) **Minimum Edge Length Convex Partitioning with Points (MELCPP):** Given a simple polygon, a set of points within its area, and a number k, is there a partitioning of its area into convex parts such that all given points reside on decomposing edges whose edge length does not exceed k?

We chose to transform MELRP from planar satisfiability (PLSAT), shown to be NP-complete in [Lich82]. MELRP transforms to MELRPP, and the proofs for MELCP and MELCPP are analogous (with slight modifications). In fact, we use a slightly less restricted version of PLSAT defined as follows: A formula $\mathcal{F}$ in conjunctive normal form with exactly 3 literals per clause (3CNF), whose variables $x_i, i = 1, \ldots, n$ form the set X and clauses $c_j, j = 1, \ldots, m$ form the set C, is said to be *planar* if the bipartite graph $G(\mathcal{F}) = (X \cup C, E)$ with $E = \{ (x_i, c_j) \mid x_i \text{ or } \bar{x}_i \text{ is a literal in } c_j \}$ is planar. PLSAT is the problem of deciding whether a given planar formula $\mathcal{F}$ is satisfiable or not.

In order to transform PLSAT to MELRP we construct a rectilinear figure $\mathcal{H}$ such that there exists a partitioning of $\mathcal{H}$ into rectangles using edge length of at most k if and only if the given planar formula $\mathcal{F}$ is satisfiable. The key idea is to lay out $\mathcal{H}$ as an image of a planar circuit such that partitionings of $\mathcal{H}$ simulate computations of the truth value of $\mathcal{F}$.

The basic component of $\mathcal{H}$ is a crooked (geniculated) wire used to transmit 0 (false) and 1 (true) signals. Figure 10 shows a piece of a wire: Its sides are square waves whose phases are the same. A wire is most efficiently partitioned into rectangles (with respect to edge length) by a series of edges either all perpendicular or all parallel to its direction (see dashed lines in the figure). We interpret a parallel partitioning as a 1, and a perpendicular one as a 0. A wire changes the value of the signal it carries if it makes a 90^0 turn, but it may split (fan-out) without changing its value; see Figure 11 for the details.

$\mathcal{H}$ is constructed as a homeomorphic image of the graph $G(\mathcal{F})$. Each vertex in X corresponds

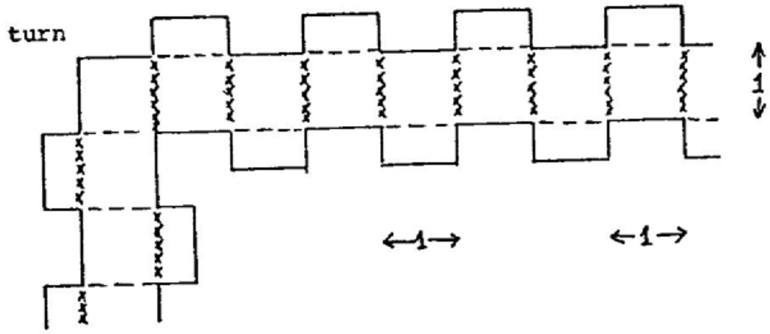


Figure 10. A piece of a crooked wire including a turn.
Two optimal decompositions are shown: - - - horizontal
xxxx vertical

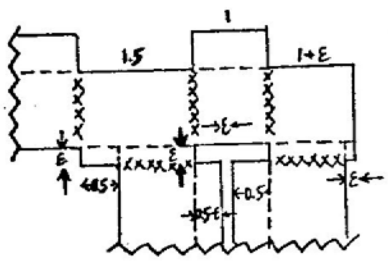


Figure 11. A wire-split; the two decompositions show a '0' and a '1'.

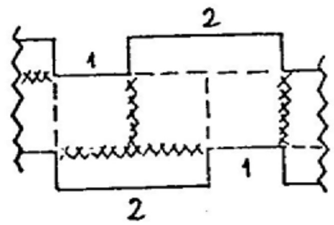


Figure 12. An inverter: a '0' (perpendicular decomposition) on the left turns into a '1' (parallel decomposition) the right and vice versa.

to a unique 90^0 turn in a wire of $\mathcal{H}$, called a *variable turn*; each vertex in C corresponds uniquely to a junction of three wires in $\mathcal{H}$, called a *clause junction*. Each edge in E is realized by a wire path between the corresponding variable turn and the clause junction. A wire is supposed to enter the clause junction carrying the truth value of the literal it corresponds to; since no distinction between x_i and $\bar{x}_i$ was made in constructing $G(\mathcal{F})$, we need to install *inverters* on the entry to a junction, as described in Figure 12. Also, wires have to enter turns and junctions with an appropriate phase for their sides, thus a *phase shifter* may be applied, as shown in Figure 13. Figure 14 shows a clause junction. Note that the pair of points P_3, P_6 is slightly not corectilinear (by ϵ); so is the pair P_3, P_{10}. Therefore, the wire entering at the bottom is narrower by $\epsilon < \frac{1}{4 \cdot \nu(\mathcal{H})}$, where $\nu(\phi)$ denotes the number of concave corners in the figure ϕ. We shall see that clause junctions behave like three-input OR gates.

Let us now cut $\mathcal{H}$ into single wires, wire-splits, inverters, phase shifters and junctions (such cuts are denoted in our pictures by ragged lines). Let $\mathcal{D}(\mathcal{H})$ denote the set of *devices* into which $\mathcal{H}$ has been cut. Given a partitioning $\mathcal{R}$ of $\mathcal{H}$ into rectangles, we associate with each edge $e \in \mathcal{R}$ the value $s_{\mathcal{R}}(e)$ to be equal to the length of e divided by the number of devices in $\mathcal{D}(\mathcal{H})$ that contain a continuous piece of e. For every $d \in \mathcal{D}(\mathcal{H})$, $\text{charge}_{\mathcal{R}}(d)$ is the sum of $s_{\mathcal{R}}(e)$ over all edges $e \in \mathcal{R}$ that have a continuous piece in d. Clearly,

$$\sum_{e \in \mathcal{R}} \text{length}(e) = \sum_{d \in \mathcal{D}(\mathcal{H})} \text{charge}_{\mathcal{R}}(d).$$

We say that a device d is *optimally charged* by a partitioning $\mathcal{R}$ of $\mathcal{H}$ if for any other partitioning, $\mathcal{Q}$,

$$\text{charge}_{\mathcal{R}}(d) - \text{charge}_{\mathcal{Q}}(d) < \epsilon \cdot \nu(d).$$

Then we have

Lemma 5. For any wire, wire-split, inverter and phase shifter d, d is optimally charged

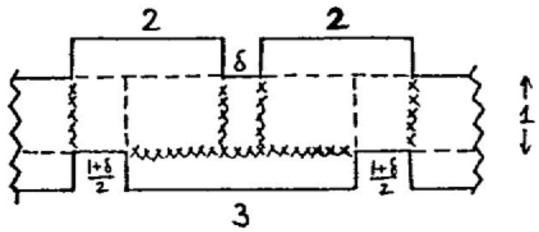
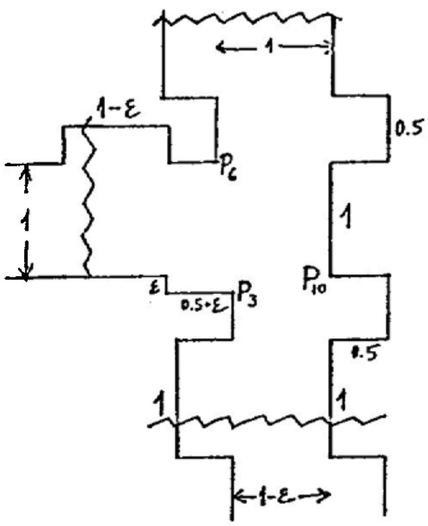
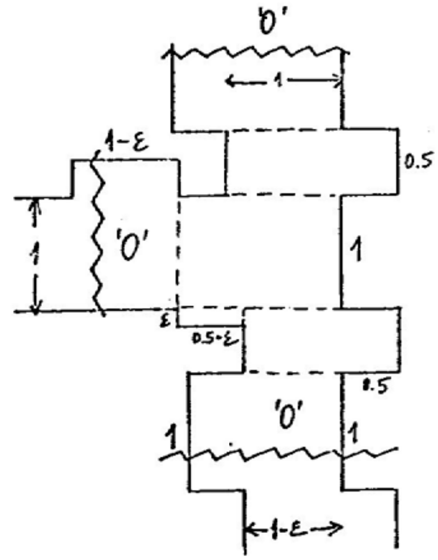


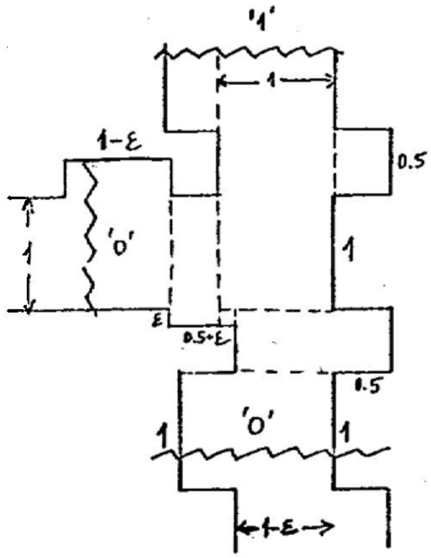
Figure 13. A phase shifter. The phase is shifted by δ.



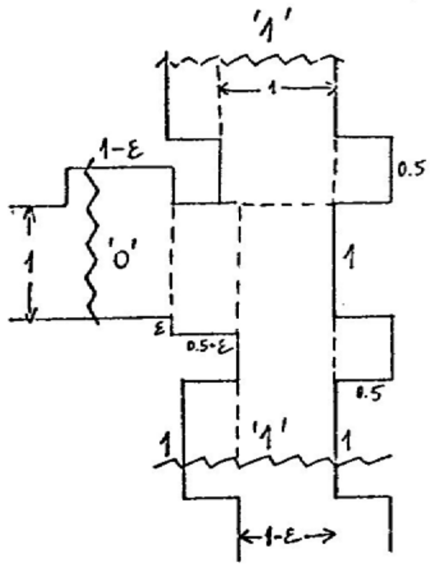
(a) Layout of the junction. Neither P_3, P_6 nor P_3, P_{10} are corectilinear.



(b) All '0' inputs yield a decomposition of length 5.5



(c) One '1' is enough to yield a decomposition of length 5+ε.



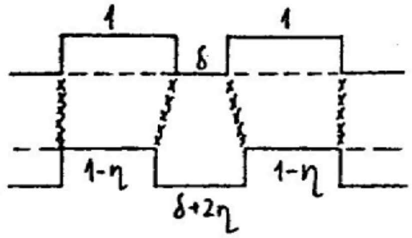
(d) Again, the decomposition length is 5+ε.

Figure 14. A clause junction and how it is being (negatively) charged.

by $\mathcal{R}$ if and only if d propagates the signal correctly. Otherwise, the charge is greater by at least $\frac{1}{2} - \epsilon \cdot \nu(d)$.

Lemma 6. If at least one wire entering a clause junction d carries a 1, then d can be charged optimally by $\mathcal{R}$ (of that wire). Otherwise, $\text{charge}_{\mathcal{R}}(d)$ is greater by at least $\frac{1}{2} - 2 \cdot \epsilon$.

Figure 15. A δ-phase-shifter for MELCD. We require $0<\delta\leq 1$ and $2(1+\sqrt{1+\eta^2})=3+2\eta+\delta$.



We say that $\mathcal{R}$ charges $\mathcal{H}$ optimally if every $d \in \mathcal{D}(\mathcal{H})$ is charged optimally by it. For each device d we can easily find the minimum possible value of $\text{charge}_{\mathcal{R}}(d)$. Let k be the sum of these minimum charges over all devices in $\mathcal{D}(\mathcal{H})$ plus $\epsilon \cdot \nu(\mathcal{H})$. From Lemmas 5 and 6 we can conclude that $\mathcal{R}$ charges $\mathcal{H}$ optimally if and only if the length of the edges in $\mathcal{R}$ is at most k. Also, by these lemmas and the construction of $\mathcal{H}$, if $\mathcal{R}$ charges $\mathcal{H}$ optimally, then the formula $\mathcal{F}$ is satisfiable.

Finally, if $\mathcal{F}$ is satisfiable then a rectangular partitioning of $\mathcal{H}$ of edge length not greater than k can be obtained by partitioning variable turns according to the satisfying truth values. Then signals are propagated along wires.

So we have

Lemma 7. $\mathcal{F}$ is satisfiable if and only if there exists a partitioning of $\mathcal{H}$ into rectangles with edge length not exceeding k.

Since the construction of $\mathcal{H}$ can be performed in logarithmic space, k and the dimensions of $\mathcal{H}$ are polynomially related to the size of $\mathcal{F}$. Also, MELRP is obviously in NP. Thus

Theorem 3. MELRP is strongly NP-complete.

Given a rectilinear figure with non-degenerate holes, we can simulate the boundaries of the holes by appropriately dense (corectilinear) points. Together with the fact that we can find an optimal rectangular partitioning of each of the holes in polynomial time (by Theorem 1), this leads to

Theorem 4. MELRP is many-one polynomially reducible to MELRPP; thus MELRPP is also (strongly) NP-complete.

To prove NP-completeness of MELCP, we slightly modify the construction of $\mathcal{H}$. Since now non-vertical and non-horizontal edges are allowed, the dimensions of the phase shifter have to be changed. Figure 15 shows the modified phase shifter. All other devices remain unchanged. Now it is more difficult to show that these devices still propagate signals correctly, since, for instance, non-vertical and non-horizontal edges may be used inside wire-splits, phase shifters, and junctions. The proof can be simplified by using a "sprout" in each junction around P_3 (in Figure 14).

The results are similar to the ones in Theorems 3 and 4:

Theorem 5. MELCP and MELCPP are strongly NP-complete.

5. Extensions and Open Problems

We have shown that the complexity of partitioning a rectilinear figure into rectangles using minimum edge length depends on whether holes are permitted or not. In general the problem is NP-complete, but when holes are disallowed the problem becomes tractable. Two major open problems remain unsolved:

(i) What if all holes are degenerate (*i.e.* they are points), but no two points are corectilinear (*i.e.* lie on the same horizontal or vertical line)?

(ii) What if, as in (i), no two points are corectilinear, and the boundary is a rectangle?

We could show neither of the problems to be NP-complete or polynomial-time solvable. Clearly, (ii) is "easier" than (i), so that resolving one in the right direction may solve them both, but on the other hand the border between P and NP may lie just between these two problems.

Other interesting extensions regarding the polynomial-time algorithm of Section 3 are the following: First, can it be generalized to the non-rectilinear case? That is, can we find a *convex* partitioning of a simple polygon using minimum edge length in polynomial time? How about triangulations that use non-boundary points? There is an interesting balance between the convexivity that led Chazelle and Dobkin to their polynomial-time algorithm for minimizing the *number* of pieces on one hand, and the Steiner tree-like property of our new minimum *edge length* criterion on the other hand. The latter took over in the rectilinear case, but the general problem remains open. Second, can the algorithm be generalized to three dimensions, where the metric will be minimum surface area? Clearly, the results of Section 4 imply that solid holes make this problem NP-complete, but it is unclear whether the added dimension in the hole-free case renders the problem intractable.

References

[AHU74] Aho,A.V., J.E. Hopcroft and J.D. Ullman, *"The Design and Analysis of Computer Algorithms"*; Addison-Wesley, Reading, Mass., 1974.

[ChDo79] Chazelle,B. and D. Dobkin, *Decomposing a Polygon into its Convex Parts*; Proceedings of the Eleventh ACM Symposium on Theory of Computation, May 1979, pp. 38-48.

[GaJh79] Garey,M.R. and D.S. Johnson, *"Computers and Intractability: A Guide to the Theory of NP-Completeness"*; W.H. Freeman and Co., San Francisco, Ca., 1979.

[GJPT78] Garey,M.R., D.S. Johnson, F.P. Preparata and R.E. Tarjan, *Triangulating a Simple Polygon*; Information Processing Letters, Vol. 7, No. 4 (June 1978), pp. 175-180.

[Lich82] Lichtenstein,D., *"Planar Formulae and their Uses"*; SIAM Journal on Computing, Vol. 11, No. 2 (May 1982), pp. 329–343.

[Lin82] Lingas,A., *The Power of Non-rectilinear Holes*; in Automata, Languages and Programmin Lecture Notes in Computer Science No. 140, pp. 369–383.

[Lin81] Lingas,A., *Heuristics for Minimum Edge Length Rectangular Decomposition*; unpublished manuscript, M.I.T., November 1981.

[LLMP79] Lodi,E., F. Luccio, C. Mugnai and L. Pagli, *On Two-dimensional Data Organization I*; Fundamenta Informaticae, Vol. 2, No. 2 (1979), pp. 211-226.

[LLMPL79] Lodi,E., F. Luccio, C. Mugnai, L. Pagli and W. Lipski, Jr., *On Two-dimensional Data Organization II*; Fundamenta Informaticae, Vol. 2, No. 3 (1979).

[Ri82] Rivest, R. L.: *The "PI" (Placement and Interconnect) System*; Proceedings of the Nineteenth Design Automation Conference, June 1982, pp. 475–481.

[ShHo76] Shamos,M.I. and D. Hoey, *Geometric Intersection Problems*; Proceedings of the Seventeenth IEEE Symposium on Foundations of Computer Science, October 1976, pp. 208-215.