AN ALGORITHM FOR CONSTRUCTING REGIONS WITH RECTANGLES: INDEPENDENCE AND MINIMUM GENERATING SETS FOR COLLECTIONS OF INTERVALS*

Deborah S. Franzblau

Daniel J. Kleitman

Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

We provide an algorithm which solves the following problem: given a polygon with edges parallel to the x and y axes, which is convex in the y direction, find a minimum size collection of rectangles, which cover the polygon and are contained within it. The algorithm is quadratic in the number of vertices of the polygon. Our method also yields a new proof of a recent duality theorem equating minimum size rectangle covers to maximum size sets of independent points in the polygon.

INTRODUCTION

The problem we address is that of constructing a planar shape with as few rectangles as possible. In other words, given a polygon, find a minimum sized collection of rectangles, which are allowed to overlap, whose union is the polygon. Here the polygon is restricted to be a union of unit squares, thus having only horizontal and vertical boundaries.

One application of this result is to the problem of efficiently creating masks for photolithography, using a pattern generator which is constrained to print rectangles. (See, for example, Mead and Conway [6], pp. 93-98, and Chaiken, et al. [1], p. 394.) Another application is that of storing and constructing pictures on a computer terminal (e.g., Masek [3]).

The general problem is known to be NP-hard. Our main result is an efficient algorithm for constructing a minimum rectangle cover when the region in question is vertically convex.

PRELIMINARY DEFINITIONS

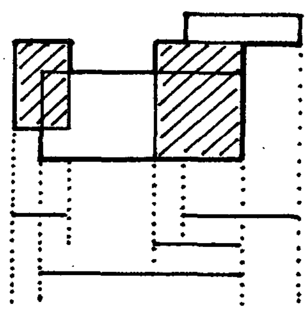
In the following, a _polygon_ is a finite subset of an infinite grid of unit squares in the plane. A polygon is _vertically_ (horizontally) _convex_, if every column (row) of squares in the polygon is connected. A _rectangle_ is a rectangular subset of unit squares, which shall be assumed to lie entirely within the given polygon. A subset of squares in a polygon is said to be _independent_ if no two squares are contained within the same rectangle. A collection of rectangles, possibly overlapping, whose union is equal to the given polygon is called a _rectangle cover_.

Previously, A. Frank noticed that for a vertically convex region, the only infor-



*Supported in part by Air Force Contract AFOSR-82-0326A

mation one needs to construct a rectangle cover is the set of distinct horizontal intervals determined by the vertical sides of the rectangles. The rectangle cover is then the set of maximal rectangles in R generated by the intervals [2]. (See Figure 1.)



A vertically convex polygon with rectangle cover and intervals representing cover indicated.

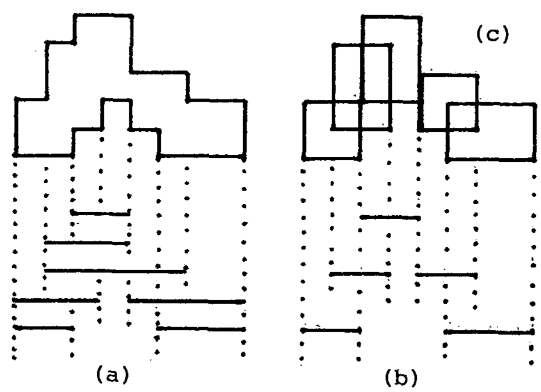
Figure 1

In a vertically convex polygon, each maximal connected subset of a row of squares determines a real interval: the projection of the row onto the real axis. Let S denote the set of all (open) intervals obtained from the polygon in this way. Then, finding a minimum rectangle cover for R can be reduced to finding a minimum size set of intervals which generate S, meaning that each interval in S is the union of generating intervals. (See Figure 2.) Conversely, an independent set of squares in R can be shown to correspond to a sequence of intervals in S such that each interval contains a point in none of its predecessors [2]. More formally, $T \subseteq S$ is <u>independent</u> if the intervals in T can be ordered $I_1, I_2, \ldots I_n$ so that each $I_k$ contains points not in $\bigcup_{j=1}^{k-1} I_j$.

When there is no such ordering, T is called <u>dependent</u>.

## CHRONOLOGY OF PREVIOUS RESULTS

| Year | Result |
|---|---|
| 1978 | Masek [3] proved that finding minimum rectangle covers for polygons is NP-hard. |
| 1979 (?) | Chvatal conjectured that, for any polygon, the size of a minimum rectangle cover is equal to the size of a maximum independent subset. Small counter-examples were constructed by Szemeredi and Chung [1]. |
| 1980 | Chaiken, et al. [1] proved Chvatal's conjecture is true, if the polygon is both vertically and horizontally convex. |




(a) Intervals determined by polygon

(b) Generating set for intervals

(c) Rectangle cover constructed from generating set.

Figure 2

1983 Györi [4] showed that the conjecture is true even if the polygon is convex in only one direction.

Here, using the interval formulation of the problem, we shall give a simple algorithm for constructing minimum generating sets, which can be transformed easily into minimum rectangle covers. The algorithm is $O(n^2)$, where n is the number of intervals, which is proportional to the number of vertices in the polygon. The basic step of the algorithm is a simple operation: replacing "chains" of overlapping intervals with the intersections of successive pairs; it is reminiscent of the augmenting path algorithm for maximum matching.

Constructing minimum generating sets and finding maximum independent sets of intervals can be viewed as dual problems. Our work provides yet another example of a combinatorial optimization problem in which such duality is the key to constructing a fast algorithm.

## CONSTRUCTING MINIMUM GENERATING SETS

We first need a nice characterization of dependent sets. It is easy to show that S is dependent if and only if S contains a subset T, such that every point in $\cup T$ (the union of intervals in T) is covered by at least two intervals in T. We call such a collection T simply dependent. We can extend our definitions to subintervals $U \subseteq \cup S$ by saying that an interval U is dependent, or simply dependent whenever the set of all intervals with both endpoints contained in U has the corresponding property. Note the important distinction between subintervals of $\cup S$ and intervals in the collection S. We call U minimal simply dependent if no proper subinterval of U is simply dependent. (See Figure 3.)

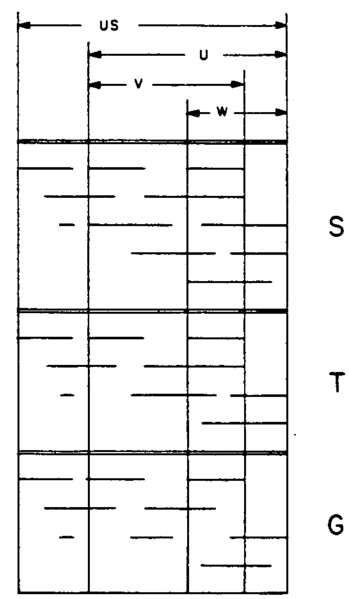
Given S, a collection of intervals such that $\cup S$ contains a simply dependent interval U, we now give a procedure which constructs a set of generators G for S with $|S| - 1$ intervals. G is the set formed by replacing consecutive pairs of maximal intervals inside U with their intersections.

### Reduction Procedure

Let T be the set of intervals in U (i.e. intervals with both ends in U). Let $\{I_1, I_2, \ldots, I_k\}$ be the maximal intervals in T ordered by increasing left endpoints.

If $k = 1$ then $G \leftarrow S - \{I_1\}$.

Else $G \leftarrow [S - \{I_1, I_2, \ldots, I_k\} \cup \{I_1 \cap I_2, I_2 \cap I_3, \ldots, I_{k-1} \cap I_k\}]$.□




S is a dependent set of intervals. U, V and W are simply dependent intervals; V and W are also minimal. T is an independent subset of S (because the left endpoints of intervals in T are strictly increasing. G is a generating set for S which is also independent. $|G| = |T|$ in this case, so G is a minimum generating set and T is a maximum independent set.

Figure 3

It is easy to see that $|G| = |S| - 1$. The simple dependence of T guarantees that each point in the interval

$$I_j - (I_{j-1} \cap I_j) - (I_j \cap I_{j+1})$$

is contained in some interval in T, which must be non-maximal, and hence contained in $I_j$. Thus, every point in $I_j$ is generated by intervals in G.

We now explain how to use the reduction to find a minimum size generating set. We first introduce some notation. If S is a collection of intervals, and $U \subseteq \cup S$, let $RUS$ be the collection of intervals obtained by applying the reduction procedure to the intervals of S contained in U.

We define another operator $D_L$, where $D_L S$ is the leftmost minimal simply dependent interval in S. To find a minimum size set of generators, we iterate the operator $RD_L$ to construct

$$RD_L S,\ (RD_L)^2 S, \ldots,\ (RD_L)^m S.$$

We stop when $(RD_L)^m S$ is independent (contains no simple dependence). It is clear that the algorithm halts since

$$|RD_L S| = |S| - 1.$$

The generating set G in Figure 3 was obtained in this manner; $m = 2$, $V = D_L S$, and $W = D_L RD_L S$.

VALIDITY OF ALGORITHM

Let

$$B_L = (D_L S,\ D_L RD_L S, \ldots,\ D_L (RD_L)^{m-1} S)$$

be the sequence of intervals reduced during the algorithm. It can be shown that each $D \in B_L$ appears only once, using several technical facts, described in the appendix.

If we define $D_R S$ to be the rightmost minimal simply dependent interval in S, then we may consider the same algorithm working from the right. If

$$B_R = (D_R S,\ D_R RD_R S, \ldots,\ D_R (RD_R)^{\ell} S)$$

then each $D \in B_R$ also appears only once by the same argument. We now state our first main result. A sketch of the proof appears in the appendix.

Theorem 1.

$$B_L(S) = B_R(S) \text{ as sets.}$$

We use Theorem 1 to show that our algorithm constructs a minimum generating set. This follows immediately from the next theorem.

Theorem 2. Let $B_L$ and m be as defined above. Let T be a maximum size independent subset in S.

Then

$$|T| = |S| - m = |S| - |B_L|.$$

Proof of Theorem 2. We prove the result by induction on $|S|$. Let

$$S_m = (RD_L)^m S,$$

the final set of generators. S is independent, so there is some point x in $\cup S$ which is covered by exactly one interval in $S_m$. Form $S' \subseteq S$ by removing all intervals in S which cross x. S' is the disjoint union of $S_1$ and $S_2$, which contain, respectively, the intervals to the right and to the left of x. Using Theorem 1, we have

$$B_L(S') = B_L(S_1) \cup B_L(S_2) = B_L(S_1) \cup B_R(S_2)$$

The key observation is that $B_L(S)$ contains exactly those intervals in $B_L(S)$ which lie strictly to the left of x, and $B_R(S_2)$ contains exactly those intervals in $B_R(S)$ (which is equal to $B_L(s)$) which lie strictly to the right of x. Further-more, if $j \geq 1$ is the number of intervals in S which contain x. then exactly $j - 1$ intervals in $B_L(S)$ contain x. This is because each application of the reduction procedure, to an interval U containing x, reduces the total number of intervals containing x by one. Thus,

$$|B_L(S')| = |B_L(S)| - (j - 1)$$

$$= m - j + 1$$

Now, if T' is an independent set in S', then clearly $T' \cup \{I\}$ is independent in S, where I is any interval containing x. Thus, the size of a maximum independent set in S is at least $|T'| + 1$. But, by induction,

$$|T'| = |S'| - |B_L(S')|, \text{ and}$$

$$1 + |T'| = 1 + |S| - j - (m - j + 1) =$$

$$|S| - m,$$

which is what we wished to show. □

## $O(N^2)$ IMPLEMENTATION

Represent intervals by ordered pairs (X, Y) of rational numbers. Assume the intervals are stored in a doubly-linked list, ordered by increasing left endpoints. In case of ties, the order is arbitrary. We also assume we have an ordered list of all endpoints, marked as to whether they are left, right, or both.

Assume that S is connected, i.e., $\cup S$ is a single interval. Otherwise find generating sets for each connected component separately.

Recall that the algorithm to find a minimum generating set has two basic steps: (1) Given S, the collection of intervals, find (L, R), the leftmost minimal simply dependent interval in $\cup S$; (2) Apply the reduction procedure to (L,R) and update S.

To find a minimal dependence quickly, we first find $R_1 < R_2 < \ldots < R_k$, all the points which are right ends of two or more intervals. For i = 1, 2, ... , we look for $(L_i, R_i)$, a minimal dependent interval with right end $R_i$, if one exists. If $R_i$ is the smallest right end such that a dependence exists, then $(L_i, R_i)$ must be the leftmost minimal dependence. It can be shown that if there is no simple dependence with right end $R_i$, then $R_i$ can be crossed off the "candidate" list permanently.

Each $(L_i, R_i)$ can be found in linear time using the following idea. Beginning at the right end $R_i$, simultaneously construct two disjoint "paths" of intervals from $R_i$. Extend paths by adding intervals. To insure that we find a minimal simple dependence, always extend the shorter path first, and always choose the interval which extends the path by the smallest non-zero amount.

The reduction procedure is easily accomplished in linear time if we observe that, if $(X_i, Y_i)$ denotes the $i^{th}$ maximal interval in (L, R), then $(X_{i+1}, Y_{i+1})$ is the leftmost interval with

$$X_i < X_{i+1} < Y_i < Y_{i+1} \leq R$$

(if we choose $Y_{i+1}$ so that $(X_{i+1}, Y_{i+1})$ is the longest interval with left end $X_{i+1}$).

Since the number of iterations of the algorithm is at most $O(n)$, where n is the number of intervals, the procedure is seen to be $O(n^2)$ in the worst case. Finally, one can show that n is proportional to m, the number of vertices in the polygon, and that constructing S and reconstructing the rectangle cover from the generating set is at worst $O(m^2)$.

## CONCLUSION

We have given a simple, efficient algorithm which finds an optimal rectangle cover, for any vertically convex, rectilinear polygon. This special case has two critical properties, which facilitate the construction of an exact algorithm. The first is one-dimensionality, which allows the reduction of a planar covering problem to an interval covering problem. The second is the duality between minimum size rectangle covers and maximum size independent sets of squares. Even in the simply-connected case (no holes in the polygon), neither of these properties are present.

It is not yet known whether our methods can be extended to give an exact algorithm for a larger class of regions. We have discovered a few special cases in which a region which is neither vertically nor horizontally convex can be split into two regions, each of which can be covered separately. In other cases, a region R may contain a vertically convex subregion which can be covered independently, then "contracted," effectively reducing R to a smaller region.

A related problem is that of partitioning an arbitrary polygon into non-overlapping, horizontal trapezoids, a problem which arises in electron beam lithography, described in [7] and [8]. For rectilinear polygons, this becomes the problem of partitioning a region into rectangles, for which there is an $O(n^{5/2})$ exact algorithm (n is again the number of vertices) [8].

Unfortunately, there exist regions which need k rectangles in a rectangle cover but which require $O(k^2)$ rectangles in a rectangle partition. So, in general, partitioning is not a good replacement for covering. Further results on the relationship between partitioning and covering, and on other heuristics for finding rectangle covers will be reported elsewhere.

## REFERENCES


[1] S. Chaiken, D.J. Kleitman, M. Saks, J. Shearer; "Covering Regions by Rectangles," SIAM J. Alg. Disc. Meth. (2), Dec. 1981, pp. 394 - 410.

[2] E. Győri; "A Minimax Theorem of New Type," 1983, to appear.

[3] W.J. Masek; "Some NP-complete set covering problems," 1978, unpublished manuscript, M.I.T.

[4] E.L. Lawler; Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.

[5] M. Aigner; Combinatorial Theory, Springer-Verlag, 1979.

[6] C. Mead, L. Conway; Introduction to VLSI Systems, Addison-Wesley, 1980.

[7] T. Asano and T. Asano; "Minimum Partition of Polygonal Regions," 24th Symposium on Foundations of Computer Science (Nov., 1983) IEEE Computer Soc., pp. 233 - 241.

[8] T. Ohtsuki; "Minimum Dissection of Rectilinear Regions," Proc. IEEE Int. Symposium on Circuits and Systems (Rome, 1982) pp. 1210 - 1213.

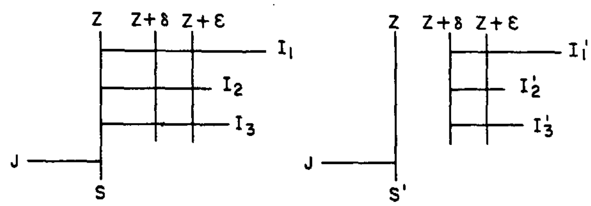

## APPENDIX (THEOREM 1)

The proof of Theorem 1 depends on several technical facts and lemmas, which we list below. Facts 1 and 2 and Lemma 1 are needed mainly to guarantee that, without loss of generality, the intervals in $B_L(S)$ are distinct, which greatly simplifies the proof. The more important Lemmas, 2 and 3, are used to transform the proof into a simple "algebraic" exercise.

Fact 1 We may assume that no point is both a left and right endpoint for intervals in S.

Fact 2 We may further assume that no point is the left endpoint of more than two intervals.

To prove Facts 1 and 2, we replace S by a new collection S', using the constructions shown in Figures A1 and A2. We then show how to construct generating sets and independent sets for S from those for S'. It can be shown that the additional processing of intervals adds only 0(n) steps to the implementation.




Construction used for Fact 1. $\varepsilon$ is chosen so that $(z, z + \varepsilon)$ contains no other endpoints of intervals, and $0 < \delta < \varepsilon$.

<u>Figure A1</u>

<u>Lemma 1</u> If $U \subseteq \cup S$ is <u>not</u> simply dependent, then applying the reduction procedure to some simply dependent interval D cannot cause U to become simply dependent.

<u>Sketch of Proof</u> Determine those points in U which must be covered by two intervals before the reduction, if U is simply dependent after the reduction. The nontrivial cases are:

$$D \subseteq U,\ U \subseteq D, \text{ and}$$

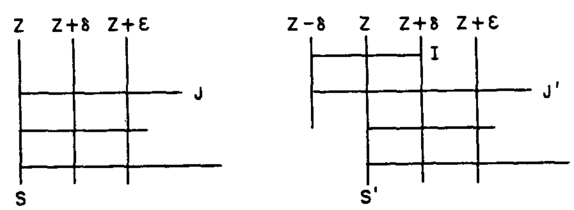
$$U \cap D \neq \emptyset (\text{but} \neq U \text{ or } D).\square$$

<u>Lemma 2</u> Let $D_1$ and $D_2$ be distinct simply dependent intervals in $\cup S$, where $D_1$ is minimal and $D_1 \nsubseteq D_2$. Then $D_2$ will remain simply dependent after the reduction procedure is applied to $D_1$.

<u>Lemma 3</u> Let $D_1$ and $D_2$ be distinct minimal simply dependent intervals. Then we can perform the reduction in either order:

$$RD_1RD_2S = RD_2RD_1S.$$

<u>Sketch of Proofs of Lemmas 2 and 3</u> The results follow trivially unless $D_1$ and $D_2$ overlap and have intervals in common. The key is that the minimality of $D_1$ insures that $D_1 \cap D_2$ is independent, which means there is a point x contained in exactly one interval inside $D_1$ and $D_2$. The simple dependence of $D_1$ and $D_2$ mean there are maximal intervals not in $D_1 \cap D_2$ which also contain x. The lemmas are established by a separate analysis of the intervals in $D_1 \cap D_2$ which are to the left of x, contain x and are to the right of x.$\square$




Construction used for Fact 2. Again, $\varepsilon$ is chosen so that $(z - \varepsilon, z + \varepsilon)$ contains no other endpoints of intervals, and $0 < \delta < \varepsilon$.

<u>Figure A2</u>

Sketch of Proof of Theorem 1

Either $D_L S = D_R S$, or

(1) Lemma 2 implies that:

$$D_L S = D_L R D_R S$$

and $D_R S = D_R R D_L S$.

(2) Lemma 3 implies that:

$$R D_L R D_R S = R D_R R D_L S.$$

To show $B_L(S) \subseteq B_R(S)$, we need only show that, given any $j$, $0 \leq j \leq m - 1$, there is some $k \geq 0$ such that:

$$D_L (R D_L)^j S = D_R (R D_R)^k S.$$

This is established by induction on $S$, using (1) and (2) above. Symmetrically,

$$B_R(S) \subseteq B_L(S). \square$$