CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING
Vol. 53, No. 2, March, pp. 132–136, 1991

Fitting Polygonal Functions to a Set of Points in the Plane*

S. L. HAKIMI

Department of Electrical Engineering and Computer Science, University of California, Davis, California 95616

AND

E. F. SCHMEICHEL

Department of Mathematics and Computer Science, San Jose State University, San Jose, California 95192

Received July 18, 1988; accepted May 9, 1990

Given a set of points $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ in $\mathbf{R}^2$ with $x_1 < x_2 < \cdots < x_N$, we want to construct a polygonal (i.e., continuous, piecewise linear) function f with a small number of corners (i.e., nondifferentiable points) which fits S well. To measure the quality of f in this regard, we employ two criteria:

(i) the number of corners in the graph of f, and

(ii) $\max_{1 \le i \le N}|y_i - f(x_i)|$ (the Chebyshev error of the fit).

We give efficient algorithms to construct a polygonal function f that minimizes (i) (resp. (ii)) under a maximum allowable value of (ii) (resp. (i)), whether or not the corners of f are constrained to be in the set S. A key tool used in designing these algorithms is a linear time algorithm to find the visibility polygon from an edge in a monotone polygon.

A variation of one of these algorithms solves the following computational geometry problem in optimal $O(N)$ time: Given N vertical segments in the plane, no two with the same abscissa, find a monotone polygonal curve with the least number of corners which intersects all the segments.

1. INTRODUCTION

Suppose we are given a set of points $S = \{P_1, P_2, \cdots, P_N\}$ in $\mathbf{R}^2$, with $P_i = (x_i, y_i)$ and $x_1 < x_2 < \cdots < x_N$. We wish to construct a polygonal (i.e., continuous, piecewise linear) function f with a small number of corners (i.e., nondifferentiable points) which fits S well. To measure the quality of f in accomplishing these goals, we will employ the following two criteria:

(i) the number of corners in the graph of f, and

(ii) $d_\infty(f, S) = \max_{(x_i, y_i) \in S} |y_i - f(x_i)|$ (the Chebyshev [illegible]

We will consider the following problems

Problem 1. Given S and $\varepsilon > 0$, construct a polygonal function f with the least number of corners such that $d_\infty(f, S) < \varepsilon$.

Problem 2. Given S and an integer $k \ge 0$, construct a polygonal function f with k or fewer corners such that $d_\infty(f, S)$ is as small as possible.

For either of these problems, we may add the restriction that the corners of the graph of f must occur in the set S. We indicate this restriction by adding S to the corresponding problem number; e.g., Problem 1S rather than Problem 1.

The purpose of this paper is to give efficient algorithms for Problems 1, 1S, 2 and 2S. The complexities of our algorithms are summarized below:

Problem 1	$O(N)$
Problem 1S	$O(N^2)$
Problem 2	$O(N^2 \log N)$
Problem 2S	$O(N^2 \log N)$.

A key tool used to attain these complexities is an efficient algorithm for computing the visibility polygon from an edge or vertex of a polygon π. (A point p is said to be visible from an edge e of π if there exists a point q on e such that the line segment pq is contained entirely in π. The visibility polygon from e is the set of points in π which are visible from e.) Imai and Iri [4] gave an $O(k)$ time algorithm to find the visibility polygon from an edge or vertex in a k-sided monotone polygon. The best time known for doing this in a general k-sided polygon is $O(k \log k)$ [1, 3, 5].

It will be apparent that the $O(N)$ algorithm for Problem 1 can be trivially modified to handle a nonuniform error bound $|y_i - f(x_i)| < \varepsilon_i$ for $i = 1, 2, \cdots, N$ with no loss in

* This work is supported by the National Science Foundation under Grants ECS-8700656 and ASC-8740647.

1049-9652/91 $3.00

asymptotic efficiency. Thus the following computational geometry problem can be solved in optimal linear time: Given N vertical segments in the plane, no two with the

with the least number of corners which intersects all the segments.

2. THE ALGORITHMS

(1) *An $O(N^2)$ Algorithm for Problem* 1S

We can describe our algorithm as follows:

Given $\varepsilon > 0$, construct a directed graph $G = G(S, \varepsilon)$ with vertex set $V(G) = \{w_1, w_2, \cdots, w_N\}$ and $(w_i, w_j) \in E(G)$ iff $i < j$ and the linear function $L_{ij}(x)$ representing the line joining the points P_i and P_j satisfies $\max_{i<k<j} |y_k - L_{ij}(x_k)| < \varepsilon$. Find a shortest path from w_1 to w_N in G, say $w_1 = w_{i_0}, w_{i_1}, \cdots, w_{i_p} = w_N$. The graph of the desired function $f(x)$ is obtained by joining P_{i_j} and $P_{i_{j+1}}$ by a straight line segment, for $j = 0, 1, \cdots, k - 1$.

Since G is acyclic, we can find a shortest path from w_1 to w_N in $O(|E|) = O(N^2)$ time. It remains to give an algo-

begin {construction of $G(S, \varepsilon)$}
  for $i = 1$ to $N - 1$ **do**
    let π_i denote the monotone polygon with vertices (in order) P_i, $(x_{i+1}, y_{i+1} + \varepsilon)$, $(x_{i+2}, y_{i+2} + \varepsilon)$, $\cdots$, $(x_N, y_N \rightarrow \varepsilon)$, $(x_N, y_N - \varepsilon)$, $(x_{N-1}, y_{N-1} - \varepsilon)$, $\cdots$, $(x_{i+1}, y_{i+1} - \varepsilon)$, P_i;
    let V_i denote the visibility polygon in π_i from the point P_i;
    $\{(w_i, w_j) \in E(G)$ if and only if $P_j \in V_i\}$
    let the x-coordinates of the vertices of V_i be $x_i = x'_1 < x'_2 < \cdots < x'_r$, and suppose that for $x'_k \le x \le x'_{k+1}$ we have $(x, y) \in V_i$ iff $l_k(x) \le y \le u_k(x)$; (see Fig. 1)
$j \leftarrow i + 1$;
$k \leftarrow 1$;
repeat
    if $x_j \le x'_k$
    then begin
      if $l_k(x_j) \le y_j \le u_k(x_j)$ **then** add (w_i, w_j) to $E(G)$;
      $j \leftarrow j + 1$
      end
    else $k \leftarrow k + 1$
  until $j > N$ or $k > r$
end {construction of $G(S, \varepsilon)$}.

It is easily verified that we require only $O(N)$ time for each i (in particular, we can construct V_i in $O(N)$ time by the algorithm of Imai and Iri [4]), and hence we can construct G in $O(N^2)$ time.

(2) *An $O(N)$ Algorithm for Problem* 1

We first need to describe the construction of a certain monotone polygon $\pi(S, \varepsilon)$. Our interest in $\pi(S, \varepsilon)$ comes

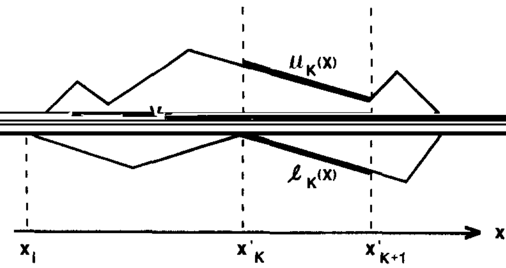

FIGURE 1

from the fact that for any polygonal function $f(x)$ with $d(f, S) < \varepsilon$, there exists a polygonal function g such that $d_\infty(g, S) < \varepsilon$, the graph of g has no more corners than the graph of f, and the graph of g lies entirely within $\pi(S, \varepsilon)$ for $x_1 \le x \le x_N$. Thus once we construct $\pi(S, \varepsilon)$, we can use the method of Iri and Imai to construct in $O(N)$ time a polygonal function $f(x)$ with as few corners as possible satisfying $d_\infty(f, S) < \varepsilon$. Since it will be obvious that our construction of $\pi(S, \varepsilon)$ takes only $O(N)$ time, this will yield the desired $O(N)$ algorithm for Problem 1.

over the interval $[x_i, x_{i+1}]$, supposing at first that $2 \le i \le N - 2$. For $x_i \le x \le x_{i+1}$, let $L^{\mathrm{Top}}(x)$ denote the linear function defined by the line joining the points $(x_{i-1}, y_{i-1} - \varepsilon)$ and $(x_i, y_i + \varepsilon)$, let $R^{\mathrm{Top}}(x)$ denote the linear function defined by the line joining the points $(x_{i+2}, y_{i+2} - \varepsilon)$ and $(x_{i+1}, y_{i+1} + \varepsilon)$, and let $M^{\mathrm{Top}}(x)$ denote the linear function defined by the line joining the points $(x_i, y_i + \varepsilon)$ and $(x_{i+1}, y_{i+1} + \varepsilon)$. We take $t(x) = \max(M^{\mathrm{Top}}(x), \min(L^{\mathrm{Top}}(x), R^{\mathrm{Top}}(x)))$ as the top boundary of $\pi(S, \varepsilon)$ on $[x_i, x_{i+1}]$. For the bottom boundary, we define the linear functions L^{Bot}, R^{Bot}, and M^{Bot} by replacing ε by $-\varepsilon$ in the definitions of L^{Top}, R^{Top}, and M^{Top}, respectively, and then take $b(x) = \min(M^{\mathrm{Bot}}(x), \max(L^{\mathrm{Bot}}(x), R^{\mathrm{Bot}}(x)))$ as the bottom boundary of $\pi(S, \varepsilon)$ on $[x_i, x_{i+1}]$. On the intervals $[x_1, x_2]$ and $[x_{N-1}, x_N]$, we take $t(x) = M^{\mathrm{Top}}(x)$ and $b(x) = M^{\mathrm{Bot}}(x)$. We illustrate $\pi(S, \varepsilon)$ in Fig. 2, in which the ith vertical segment has length 2ε with P_i as midpoint. Note that the set $\{(x, y) | x_i \le x \le x_{i+1}$ and $b(x) \le y \le t(x)\}$ is convex for every i. For each value of i we can construct $t(x)$ and $b(x)$ in $O(N)$ time, and hence $\pi(S, \varepsilon)$ can be constructed in $O(N)$ time.

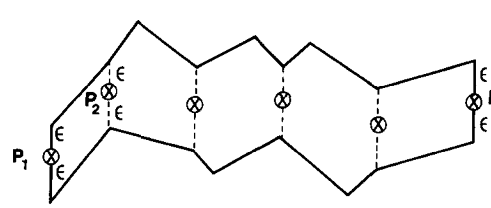

FIG. 2. $\pi(S, \varepsilon)$.

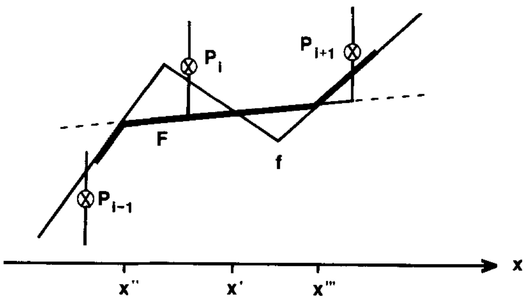

FIGURE 3

We now prove the following lemma.

LEMMA. *Let f be a polygonal function with $d_\infty(f, S) < \varepsilon$. Then there exists a polygonal function g which satisfies*

(*i*) $b(x) \le g(x) \le t(x)$ *for* $x_1 \le x \le x_N$

(*i.e., the graph of g lies entirely within* $\pi(S, \varepsilon)$ *for* $x_1 \le x \le x_N$*; in particular,* $d_\infty(g, S) \le \varepsilon$);

and

(*ii*) *the graph of g has no more corners than the graph of f.*

Proof. If

$$b(x) \le f(x) \le t(x) \quad (1)$$

holds for $x_1 \le x \le x_N$, we take $g = f$. Otherwise, let x' be the smallest abscissa in (x_1, x_N) at which the graph of f leaves $\pi(S, \varepsilon)$; i.e., (1) holds for $x_1 \le x \le x'$ but fails for $x' < x < x + \delta$ if $\delta > 0$ is sufficiently small. Suppose that $x_i \le x' < x_{i+1}$. We will show how to alter the graph of f to obtain the graph of a new function F satisfying $d_\infty(F, S) < \varepsilon$ which remains inside $\pi(S, \varepsilon)$ for $x_1 \le x \le x_{i+1}$ and [illegible] then iterate this alteration procedure until we obtain the desired function g.

We assume that for any i, $1 \le i \le N - 1$, the graph of f has at most one corner in (x_i, x_{i+1}), since otherwise we can first alter the graph of f as follows: Replace the portion of the graph on an offending interval (x_i, x_{i+1}) by the line segment joining $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$.

Without loss of generality, let us assume the graph of f first leaves $\pi(S, \varepsilon)$ on the lower boundary of $\pi(S, \varepsilon)$. We now consider two cases depending on the form of $b(x)$ on the interval $[x_i, x_{i+1}]$.

Case 1. $b(x) = M^{\text{Bot}}(x)$ for $x_i \le x \le x_{i+1}$. If $R^{\text{Bot}}(x) \ge M^{\text{Bot}}(x)$ on $[x_i, x_{i+1}]$, we first extend the graph of $M^{\text{Bot}}(x)$ to the right of x_{i+1} until it first intersects the graph of f at abscissa $x'' \ge x_{i+1}$. We then replace the portion of the graph of f on the interval $[x', x'']$ by the graph of $M^{\text{Bot}}(x)$ on $[x', x'']$ to obtain the graph of F. The resulting function F satisfies $b(x) \le F(x) \le t(x)$ for $x_1 \le x \le x_{i+1}$ and the graph of F has no more corners than the graph of f.

Suppose next that $L^{\text{Bot}}(x) \ge M^{\text{Bot}}(x)$ on $[x_i, x_{i+1}]$. Extend $M^{\text{Bot}}(x)$ to the left beyond x_i until it first intersects the graph of f at abscissa $x'' \in (x_{i-1}, x_i)$. There is also an abscissa $x''' > x'$ in $(x_i, x_{i+1}]$ at which the graphs of f and M^{Bot} intersect. We replace the portion of the graph of f on $[x'', x''']$ by the graph of $M^{\text{Bot}}(x)$ on $[x'', x''']$ to obtain the graph of F (see Fig. 3). It is easy to see than F has no more corners than F. Moreover, $b(x) \le F(x) \le t(x)$ for $x_1 \le x \le x_{i+1}$. (For $x'' \le x \le x_i$, this follows from the convexity of $\pi(S, \varepsilon)$ on $[x_{i-1}, x_i]$ and the fact that $(x'', f(x)) \in \pi(S, \varepsilon)$.)

Case 2. $b(x) \ne M^{\text{Bot}}(x)$ for $x_i \le x \le x_{i+1}$. Let $x'' > x'$ denote the abscissa in $(x_i, x_{i+1}]$ at which the graphs of f and b intersect, and let $x_{\min}$ denote the abscissa at which $b(x)$ attains its minimum value in $[x_i, x_{i+1}]$. We have three possibilities.

Subcase 2(*i*). $x', x'' \le x_{\min}$. Extend the graph of L^{Bot} to the left beyond x_i until it intersects the graph of f at abscissa $x''' \in (x_{i-1}, x_i)$. Arguing as in the previous case, we can replace the portion of the graph of f on (x''', x'') by the graph of L^{Bot} to get the graph of a function F with the desired properties.

Subcase 2(*ii*). $x', x'' \ge x_{\min}$. Extend the graph of R^{Bot} to the right beyond x_{i+1} until it meets the graph of f at [illegible] $[x', x''']$ to get the graph of the function F with the desired properties (see Fig. 4).

Subcase 2(*iii*). $x' < x_{\min} < x''$. We then extend L^{Bot} to the left beyond x_i until it meets the graph of f at abscissa x''' in (x_{i-1}, x_i) (resp, extend R^{Bot} to the right beyond x_{i+1} until it meets the graph of f at abscissa x'''' in (x_{i+1}, x_{i+2})). [illegible] portion of the graph of L^{Bot} (resp, R^{Bot}) on $[x''', x_{\min}]$ (resp, $[x_{\min}, x'''']$) to obtain the graph of the function F with the desired properties.

This completes the proof of the lemma. ■

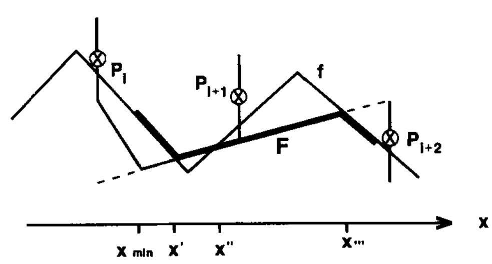

FIGURE 4

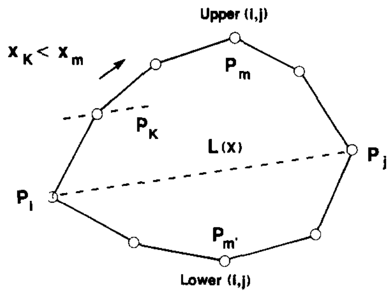

FIG. 5. The convex hull M_{ij}.

(3) *$O(N^2 \log N)$ Algorithms for Problems* 2 *and* 2S

Both algorithms are quite similar. Given k, the upper bound on the number of corners, we first develop a list of candidates for the best possible corresponding ε (there will be $O(N^2)$ candidates), and then sort this list of candidates in $O(N^2 \log N)$ time. Note that for any $\varepsilon > 0$, we can determine the corresponding smallest number of corners $k(\varepsilon)$ required using the algorithm for Problem 1 or 1S; moreover if $\varepsilon_1 < \varepsilon_2$ then $k(\varepsilon_1) \geq k(\varepsilon_2)$. Thus we can search the sorted list of ε's using binary search to find the smallest candidate ε which allows us to use just k or fewer corners. During this search we will consider $O(\log N)$ candidate ε's at a maximum cost of $O(N^2)$ per candidate considered (the time for the algorithm for Problem 1 or 1S). Thus it suffices to show that we can develop the list of candidate ε's in $O(N^2 \log N)$ time.

To develop the list for either problem, we consider all subsets of three or more consecutively indexed points in S (i.e., all $S_{ij} = \{P_i, P_{i+1}, \cdots, P_j\}$ for $j - i \geq 2$) in lexiographic order $S_{1,3}, \cdots, S_{1,N}, S_{2,4}, \cdots, S_{2,N}, \cdots, S_{N-2,N}$. We will obtain a candidate ε for each S_{ij} in $O(\log N)$ updating time (as we will show). Since there are $O(N^2)$ S_{ij}'s, the total time to develop the list of ε's will be $O(N^2 \log N)$.

Consider first Problem 2S. For any S_{ij}, first compute its convex hull H_{ij} either directly (if $j - i = 2$) or by updating the hull $H_{i,j-1}$ (if $j - i > 2$). It is well known [8] that this can be done in $O(\log N)$ time. Note that P_i and P_j divide the boundary of the hull into two paths which we will call Upper(i, j) and Lower(i, j). Let $L(x)$ denote the linear function whose graph joins P_i and P_j. We seek a point $P_m \in S$ on Upper(i, j) whose vertical distance to $L(x)$ is as large as possible. Note that for any point P_k on Up- [illegible] considering a line segment parallel to the graph of $L(x)$ through P_k (see Fig. 5 which illustrates the case $x_k < x_m$. Thus we can use binary search to find P_m in $O(\log N)$ time. We next find an analogous extreme point $P_{m'}$ on Lower(i, j) in $O(\log N)$ time, and take $e_{i,j} = \max\{|L(x_m) - y_m|, |L(x_{m'}) - y_{m'}|\}$ as the candidate ε for S_{ij}. It is immediate that for any k, the smallest corresponding ε must equal ε_{ij} for some set S_{ij}.

For Problem 2, consider any polygonal function f with at most k corners and with $d_\infty(f, S) = \varepsilon$ as small as possible. It is easy to verify that there exists a consecutive subset S_{ij} with the following properties:

(i) The convex hull H_{ij} is supported by two parallel lines l_1, l_2 which are vertical distance 2ε apart, with one of the lines, say l_1, passing through two consecutive points P_l, P_m on the hull, and the other line l_2 passing through a point P_k on the hull with x_k between x_l and x_m; and

(ii) the graph of f on $[x_i, x_j]$ is a straight line parallel to l_1 and l_2 and vertically halfway between (see Fig. 6).

(If not, we could slightly perturb various segments of the graph of f to obtain the graph of a new polygonal function F whose graph has at most k corners and satisfies $d_\infty(F, S) < \varepsilon$, a contradiction.)

This suggests that the candidate ε for S_{ij} can be obtained as follows: Take the convex hull H_{ij}, and find the maximum distance ε_{ij} from a point P_k on the hull to the point on the hull vertically opposite P_k. Half this distance is the desired ε_{ij}.

We now describe a simple updating procedure which allows us to obtain ε_{ij} in $O(\log N)$ time. If $j - i = 2$, compute H_{ij} and ε_{ij} directly. Otherwise obtain H_{ij} by updating $H_{i,j-1}$ (i.e., by finding points P_k, P_e on $H_{i,j-1}$ such that the lines from P_j to P_k and P_l are support lines for $H_{i,j-1}$). Without loss of generality, assume $x_k < x_l$. Starting at P_l move along the boundary of $H_{i,j}$ in the direction of decreasing x until we encounter two consecutive points P_a, P_b on the hull with $x_b \leq x_k < x_a$ (see Fig. 7). We can find P_a in $O(\log N)$ time using binary search. Then determine the maximum vertical distance from any of the points $P_l, \cdots, P_a$ (darkened in Fig. 7) to the line segment joining P_j and P_k; we can do this in $O(\log N)$ time using a binary searching method analogous to the one for Problem 2S. If $\varepsilon' > \varepsilon_{i,j-1}$, we take $\varepsilon_{ij} = \varepsilon'$; otherwise we take $\varepsilon_{ij} = \varepsilon_{i,j-1}$. This ε_{ij} is the candidate ε corresponding to S_{ij}. This completes the description of the algorithms for Problems 2 and 2S.

We conclude with two remarks about Problem 2. For $k = 0$, Problem 2 can be done in $O(N)$ time using a recent

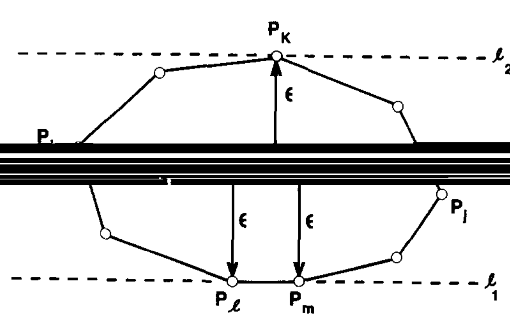

FIGURE 6

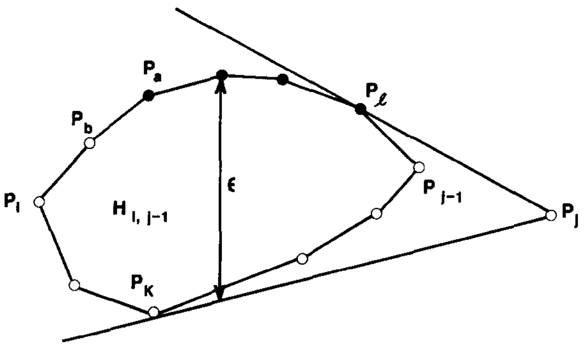

FIGURE 7

result of Dyer [2] and Megiddo [6]. To find the linear function $f(x) = ax + b$ with $d_\infty(f, S)$ as small as possible, we need only solve a linear programming problem in the three variables a, b, and D:

$$\begin{cases} \min D \\ \text{s.t.} \quad |y_i - (ax_i + b)| \leq D, \text{ for } i = 1, 2, \cdots, N. \end{cases}$$

The N given constraints can be converted to $2N$ linear constraints, and thus we can find a, b in $O(N)$ time by the result of Dyer and Megiddo. A second remark is that our algorithm for Problem 2 gives the least ε for *all* the values $k = 0, 1, \cdots, N - 2$ in $O(N^2 \log N)$ time. One might expect a better time (e.g., $O(N \log N)$) to find the optimal ε corresponding to a *single* value of k. Unfortunately, our method requires $O(N^2 \log N)$ time just to [redacted] candidates even for a single value of k.

3. RELATED PROBLEMS

Imai and Iri [4] considered a problem closely related to the ones considered above. Suppose we are given a set $S = \{P_1, \cdots, P_N\}$ of points in $\mathbf{R}^2$ with $x_1 < x_2 < \cdots < x_N$. Consider the polygonal function f_S whose graph is obtained by joining the points in S in order of increasing abscissa; the graph of f_S has $N - 2$ corners. Suppose we wish to approximate f_S by another polygonal function f with a smaller number of corners. The Chebyshev error of the approximation is $d_\infty(f, f_S) = \max_{x_1 \leq x \leq x_N} |f(x) - f_S(x)|$.

There are obvious analogues of the four earlier problems. For the analogues of Problems 1S and 2S, the functions constructed above would clearly be optimal approximations here as well. Imai and Iri [4] gave an $O(N)$ algorithm for the analogue of Problem 1, and raised the question of the complexity of the analogue of Problem 2. Unfortunately it does not seem that our approach to Problem 2 could work for their analogue since it can be shown that finding the candidates for ε (given k) may require finding the exact roots of a general polynomial of degree k. On the other hand, one can approximate the best ε corresponding to k to b bits of accuracy in $O(bN)$ time by using the $O(N)$ algorithm for Problem 1 in an obvious way.

One may also wish to fit a *discontinuous* piecewise linear function to a set of points in $\mathbf{R}^2$. In fact, the algorithms which suggest themselves are completely analogous to the ones developed here for the continuous case, and the complexities appear to be no better.

Finally we note that all the algorithms discussed thus far are off line. Certainly it would be interesting to develop efficient on-line algorithms for Problems 1 and 1S. O'Rourke [7] has given an optimal $O(N)$ on-line algorithm for a problem somewhat related to Problem 1 and 1S; i.e., the problem of piercing N data ranges by a *discontinuous* piecewise linear function having as few segments as possible. Although O'Rourke's solution can be modified to give a solution to Problem 1 (which requires a *continuous* piercing function) on the same off-line input data in certain simple instances, it does not appear that O'Rourke's solution can be modified in any reasonable way to yield a corresponding solution for Problem 1 in general.

REFERENCES

1. B. Chazelle and L. J. Guibas, Visibility and intersection problems in plane geometry, in *Proc. First ACM Sympos. on Computational Geometry, 1985*, 135–146.
2. M. E. Dyer, Linear time algorithms for two- and three-variable linear programs, *SIAM J. Comput.* **13**, 1984, 31–45.
3. H. El Gindy, An efficient algorithm for computing the weak visibility polygon from an edge in simple polygons, unpublished note, McGill University, 1984.
4. H. Imai and M. Iri, An optimal algorithm for approximating a piecewise linear function, *J. Information Processing* **9**, 1986, 159–162.
5. D. T. Lee and A. Lin, Computing the visibility polygon from an edge, *Comput. Vision Graphics Image Process.* **34**, 1986, 1–19.
6. N. Megiddo, Linear time algorithm for linear programming in $\mathbf{R}^3$ and related problems, *SIAM J. Comput.* **12** (1983), 759–776.
7. J. O'Rourke, An on-line algorithm for fitting straight lines between data ranges, *Comm. ACM* **24**, 1981, 574–578.
8. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.