Dilation-Free Graphs in the l_1 Metric

J. Cáceres
Dpto. de Estadística y Matemática Aplicada, Universidad de Almería, 04120 Almería, Spain

C. I. Grima, A. Márquez, and A. Moreno-González
Dpto. de Matemática Aplicada I, Universidad de Sevilla, 41012 Seville, Spain

The dilation-free graph of a planar point set S is a graph that spans S in such a way that the distance between two points in the graph is no longer than their planar distance. Metrically speaking, those graphs are equivalent to complete graphs; however they have far fewer edges when considering the Manhattan distance (we give here an upper bound on the number of saved edges). This article provides several theoretical, algorithmic, and complexity features of dilation-free graphs in the l_1-metric, giving several construction algorithms and proving some of their properties. Moreover, special attention is paid to the planar case due to its applications in the design of printed circuit boards. **NETWORKS, Vol. 49(2), 168–174 2007**

Keywords: dilation; Manhattan distance; complete geometric graph; planar graphs

1. INTRODUCTION

Given a geometric graph G, that is, a graph whose vertices have fixed coordinates and whose edges are straight-line segments, its *dilation* is the maximal ratio of the length of the shortest path between two vertices and their geometric distance. For communication or transportation networks, the dilation tells us exactly how much longer it is to go through the network rather than from one vertex to another on a straight line. A network spanning an n-point set S with dilation one and having the minimum number of edges will be called the *dilation-free graph* of S, denoted by $M_n(S)$ or simply M_n. This graph is trivially well defined when considered in the l_2-metric, the Euclidean distance. Later in this article we prove that M_n is also unique when the l_1-metric (the Manhattan distance) is used.

When dealing with Euclidean distance, the dilation-free graph of points in general position is the complete graph. Despite this fact, a number of intermediate and acceptable alternatives have been proposed in the literature (see, i.e., [6–8, 10]). Furthermore, the planar case has attracted attention as well, and thus in [1] planar graphs with no more than $O(n)$ edges and a dilation less than $\sqrt{10}$ are given. Finally, other authors have studied the problem for certain subsets of graphs (a survey of these results can be found in [5]).

However, not every practical situation involves the Euclidean distance, as occurs in many applications in Computational Geometry [9] or in the design of printed circuit boards. In this latter case, it is more appropriate to use the l_1-metric (or Manhattan distance) because the problem is usually posed as how to connect a set of terminals in a circuit using the shortest set of isothetic-drawn wires, that is, the wires have to be parallel to the axes. We will prove that given a point set S, the graph $M_n(S)$ contains far fewer edges than the complete graph, and therefore is the best connection layout for those terminals. Figure 1 shows a graphical example that compares the number of edges of the dilation-free graphs on the same point set using the Euclidean and the Manhattan distances.

This article is organized as follows. Section 2 takes a closer look and explores the properties of M_n, and in addition, shows two different algorithms for constructing it. In Section 3, planar dilation-free graphs are considered, producing a result that characterizes when they exist and that can be easily implemented. Also, it provides a simpler construction algorithm for the planar case than the general one. Finally, we present our conclusions in Section 4.

2. PROPERTIES AND ALGORITHMS

The main results of this article are established in this section, which is subdivided into two parts. In the first subsection it is shown that dilation-free graphs have strictly fewer edges than the complete graph, which is the most important feature of the kind of distance we deal with, the Manhattan distance. In the second subsection, two algorithms for constructing M_n are given.

Received July 2005; accepted August 2006
Correspondence to: J. Cáceres; e-mail: jcaceres@ual.es
Contract grant sponsors: FQM 164, FQM 305, HUM 348 and BFM 2001-2474
DOI 10.1002/net.20152
Published online in Wiley InterScience (www.interscience.wiley.com).

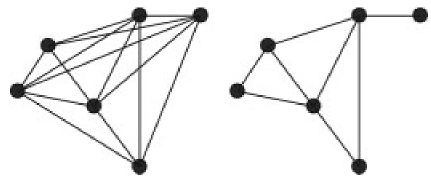

FIG. 1. Two dilation-free graphs on the same point set using the Euclidean and the Manhattan distances (for the sake of simplicity, the edges of M_n are marked as straight-line segments).

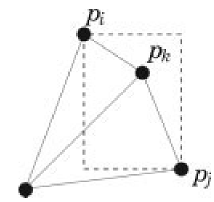

FIG. 2. Geometric proof of Lemma 1.

First, a simple yet useful result is proved to characterize whether or not two points are adjacent in M_n. Here and subsequently, x_i and y_i denote the abscissa and ordinate of a given point p_i, an edge of M_n connecting two points p_i and p_j is written as p_ip_j, and a path joining p_i and p_j through the points $q_1, q_2, \dots, q_n$ is shortened to $p_iq_1q_2\dots q_np_j$. The reader should refer to [3] for additional graph-theoretic notation.

Lemma 1. *Let p_i and p_j be two points in a point set S. Then p_ip_j is an edge of $M_n(S)$ if and only if the smallest-area isothetic rectangle enclosing p_i and p_j contains no other point in S.*

Proof. If the point p_k lies in this aforementioned rectangle, then the path $p_ip_kp_j$ has the same length as p_ip_j and consequently p_ip_j is not an edge of M_n (see Fig. 2). Conversely, if the rectangle does not contain any other point in S, the length of any path connecting p_i and p_j is greater than its Manhattan distance, and therefore, p_ip_j is in M_n. ∎

Owing to this lemma, the dilation-free graph of a point set is well defined, because deciding whether two points are adjacent in $M_n(S)$ solely depends on their coordinates.

2.1. Dilation-Free Graphs and Complete Graphs Are Distant Apart

The aim of this subsection is to state that, contrary to the Euclidean case, dilation-free graphs in the l_1-metric with at least five points contain strictly fewer edges than the complete graph. Besides that, we will also quantify this difference.

Denoting $|G|$ as the number of edges of G and K_n as the complete graph on n vertices, the first assertion can be reformulated as follows.

Lemma 2. *For every point set S such that $|S| = n \geq 5$, we have $|M_n(S)| < |K_n|$.*

Proof. Consider the smallest isothetic rectangle that encloses S. By definition, every side of this rectangle is determined by at least one point of S, so if $|S| \geq 5$ then it has an interior vertex or two vertices on the same side of the rectangle. In either case, it is possible to find a vertex that lies on the shortest path between two other points. ∎

Note that although the graph $M_n(S)$ has fewer edges than K_n, they are metrically equivalent, that is, both of them contain the same information about distances among points. A natural question that arises now is how much the graphs differ, or more explicitly, if there exists an upper bound on $|K_n - M_n|$ when n grows. The next result gives such a bound.

Theorem 3. *Let S be a set of n points. Then $|K_n - M_n| \in \Theta(n^2)$.*

Proof. Suppose that M_n contains $c\binom{n}{2}$ edges. Then the subgraph induced by a randomly chosen subset of five points would have, in expectation, $10c$ edges. However by Lemma 2, all five-element subsets induce graphs with at most nine edges, so $c \leq \frac{9}{10}$ and $|K_n - M_n| \geq \frac{1}{10}\binom{n}{2}$ for $n \geq 5$. ∎

An important point to note here is that despite the above result, dilation-free graphs may have a quadratic number of edges. Consider, for example, the graph of Figure 3 where the points are placed in convex position and every wedge contains a quarter of them. In this case, points lying in opposite wedges are joined in M_n so the number of edges is quadratic.

2.2. Computational Construction

Having established the minimum number of edges of $K_n - M_n$, we devote the rest of this section to the computational construction of M_n. We propose two algorithms: the first one, which runs in optimal $O(n \log n + m)$ time, where m is the number of edges of M_n, and a preprocessing method, which decides in time $O(\log n)$ whether or not an edge of the complete graph belongs to M_n. This preprocessing algorithm runs in time $O(n \log n)$ in the worst case.

Note that Lemma 1 provides a "brute force" approach, which can be carried out in time $O(n^3)$. Every edge of M_n can be computed by checking if the smallest-area isothetic rectangle defined by its extremes contains any other point.

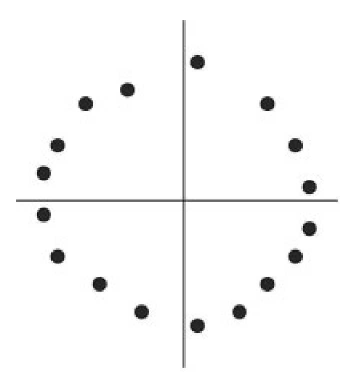

FIG. 3. In the worst case, M_n has $O(n^2)$ edges.

Our first algorithm, running in optimal time $O(n \log n + m)$, uses a line sweep approach: For each point p to the left of the sweep line, $a(p)$ and $b(p)$ will denote the points above and below p, respectively, having ordinates closest to that of p in the vertical slab bounded on the left by p and on the right by the sweep line (see Fig. 4 where the pointers a and b have been represented as arrows). Then, whenever the sweep line crosses p, a single binary search will place it within the list of points to the left of the sweep line sorted by ordinate. The neighbors of p in M_n will be found by following chains of a and b pointers from the points next to p in that sorted list, and for a neighbor of p, its a or b pointer will be redirected to p. Consider, for instance, the point p_5 in Figure 4. Once the point is inserted, the pointer $a(p_3)$ is redirected to p_5, and all the previous points, joined to p_3 by a chain of b pointers, are joined to p_5.

ALGORITHM DFG
let $P := \{p_1, p_2, \dots, p_n\}$ be the points ordered from left to right;
set $L := \{-\infty, +\infty\}$;
set $a(p_i) := +\infty$ and $b(p_i) := -\infty$ for all $i = 1, \dots, n$;
let M_n be the graph with P as vertex set and no edges;
for $i := 1$ **to** n **do**
    insert p_i in L;
    let r and s be the points next to p_i in L at its left and right;
    while $r \neq -\infty$ **do**
        add rp_i to M_n;
        $a(r) := p_i$;
        $r := b(r)$;
    while $s \neq +\infty$ **do**
        add sp_i to M_n;
        $b(s) := p_i$;
        $s := a(s)$;
return M_n as the output.

Theorem 4. *Algorithm DFG constructs the dilation-free graph of a n-point set in optimal time $\Theta(n \log n + m)$, where m is the number of edges.*

Proof. Primarily, it will be shown that every edge constructed by DFG is in M_n. If p_ip_j is one of these edges, then we may assume that $i < j$ without loss of generality and that $y_i < y_j$, because the other case is symmetrical. The edge p_ip_j is built by the algorithm just when the sweeping line is on p_j; thus, we will freeze the values of pointers a and b, and the list L to those they have at that moment.

If p_i is next to p_j in L then it has the closest ordinate to p_j below it. Thus, the isothetic rectangle defined by both points is empty and by Lemma 1, $p_ip_j \in M_n$.

On the other hand, we claim that if qp_j is in M_n and $b(q) \neq -\infty$, then $b(q)p_j$ is also an edge of M_n. To show this, divide the smallest area isothetic rectangle R defined by $b(q)$ and p_j into two rectangles: R_1, which is the intersection of R with the rectangle given by p_i and p_j, and $R_2 = R \backslash R_1$. As we have just seen in the previous paragraph, R_1 does not contain interior points. Moreover, no point lies inside R_2, because this contradicts that p_j has the closest ordinate to q below it and to the left of the sweeping line. Hence, R contains no points and $b(q)p_j$ is in M_n by Lemma 1.

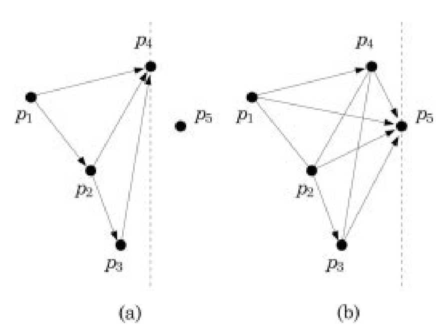

FIG. 4. During the execution of DFG, a new point is included in M_n.

Reciprocally, suppose now that the edge $p_ip_j \in M_n$ is not built by DFG. We can assume that both points are not adjacent in L; hence, let p_k be the vertex adjacent to p_j and below it. Certainly, $k < i$ because, to the contrary, p_k will lie inside the rectangle defined by p_i and p_j, and this contradicts that p_ip_j is in M_n.

This means that $a(p_i) \neq p_j$, but there exists a point p_k with $k < j$ such that $a(p_k) = p_j$. Moreover, $k < i$ holds, because, to the contrary, p_k will be contained in the rectangle defined by p_i and p_j, which contradicts that p_ip_j is an edge of M_n. Beginning with p_k and following successively the pointer b, we can construct a sequence of points $q_1q_2 \dots q_t$, where $b(p_k) = q_1, b(q_i) = q_{i+1}, \forall i \in \{1, \dots, t-1\}$ and $b(q_t) = -\infty$. If p_i is one of the points of this sequence then p_ip_j will be constructed by DFG and the proof is complete. Let us show that statement.

Suppose the plane is divided into four isothetic wedges centered at p_i. Then the northeastern wedge is free of points of this sequence because, to the contrary, p_ip_j will not be an edge of M_n.

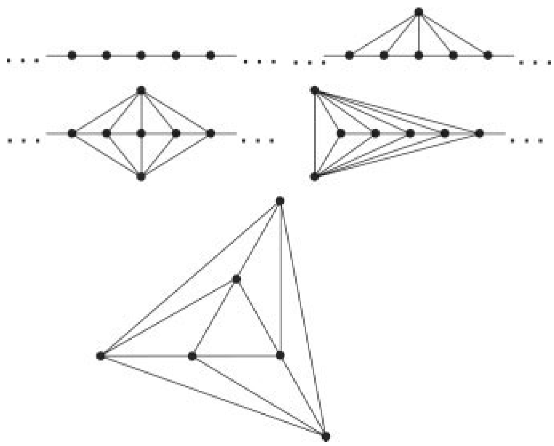

FIG. 5. Planar dilation-free graphs for the Euclidean distance.

On the other hand, suppose the sequence begins at the northwestern wedge. Assume for the time being that it also ends there, then certainly $b(q_t) = p_i$. On the contrary, if it follows to the southwestern wedge and q_i is the last point in the northwestern one, then $b(q_i) = p_i$ and so $q_{i+1} = p_i$. Finally, if the sequence "jumps" to the southeastern wedge, again we have $b(q_i) = p_i$, where q_i is the last point to the northwest of p_i.

Finally, consider the algorithm's running time in the worst case. It is clear that a point insertion in the sorted list L can be done in $O(\log n)$ time. On the other hand, notice that every step of following a pointer a or b can be viewed as moving from one extreme of an edge of M_n to another. Because after using it the pointer is redirected, one can consider that every edge is used at most once. Hence, the algorithm's running time is $\Theta(n \log n + m)$. ∎

Again, Lemma 1 suggests a new algorithm for checking whether or not a pair of points p_i and p_j are joined in M_n. From that result it clearly follows that p_i and p_j are not adjacent if and only if there exists a third point p_k in the enclosing rectangle containing p_i and p_j. From this, the idea of the algorithm is to check for any of these rectangles whether or not contain another point.

This task can be efficiently done by arranging the points in a layered range tree (for details, we refer the reader to [2]), which is previously constructed in $O(n \log n)$ time. Once we have this tree, it only remains to query repeatedly about every rectangle in the way which was described.

ALGORITHM EDGE TEST
let $S := \{p_1, p_2, \ldots, p_n\}$ be the set of points;
let G be the graph having vertex set S and no edges;
let T be the layered tree of S;
let p_i and p_j be a pair of points;
make a query in the rectangle defined by p_i and p_j;
if the rectangle is empty
    then answer "$p_i p_j$ is an edge of $M_n(S)$"
    else answer "$p_i p_j$ is not an edge of $M_n(S)$"

Theorem 5. *Given a set of points S, Algorithm EDGE-TEST checks if two points are joined in $M_n(S)$. This can be done in $O(\log n)$ time.*

Proof. The correctness of the algorithm comes from Lemma 1.

On the other hand, the query time in a layered range tree is known to be $O(\log n + k)$ where k is the number of points reported in the query range [2]. For our purposes, it is only interesting to know whether or not the range is empty so the constant k can be dismissed. ∎

3. PLANAR DILATION-FREE GRAPHS

In the design of land transportation networks, it is desirable for a network to have not only minimal dilation, but also to contain no crossing lines. This and many other applications lead us to study dilation-free graphs where the planar restriction is imposed. Turning our attention to the Euclidean distance once again, we have come across a surprising result about this issue in the literature [4]: namely, exactly four infinite families of planar dilation-free graphs exist, plus a single one (see Fig. 5).

The same problem can be posed using the Manhattan distance, because most of the uses of the l_1-metric concern the design and construction of printed circuit boards where the wires are not allowed to intersect except at a terminal. In this section, we discuss the conditions under which the dilation-free graph of a point set may be planar, and we present an algorithm for testing such conditions. Finally, we give another algorithm, simpler than DFG, for constructing the planar dilation-free graph on a point set whenever it is possible.

We say that $M_n(S)$ is *geometrically planar* or simply *planar*, if no crossing edges exist in the straight-line drawing of $M_n(S)$. Four points p_i, p_j, p_k and p_l in S are said to form an *empty quadrilateral* if there exists a positive area isothetic rectangle such that no point in S lies in its interior and every side of the rectangle contains exactly one of the those four points.

A key concept for the rest of the section will be the *top-strip* of a point. Let p_i be a point in S and let p_j and p_k be

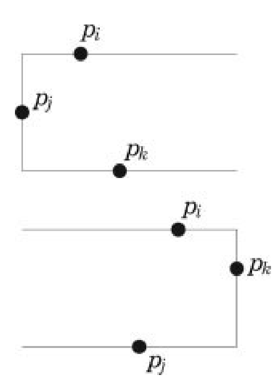

FIG. 6. Two alternatives for the top-strip $\overline{p_i}$.

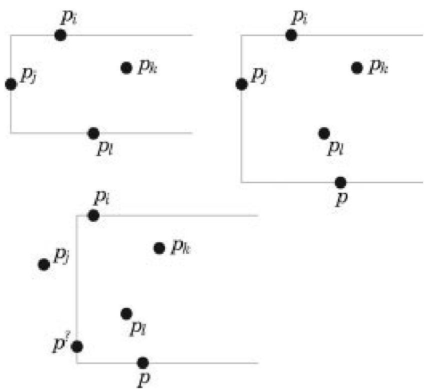

FIG. 7. Cases considered in Theorem 6.

the points below it with closest abscissa to its left and to its right (see Fig. 6). Suppose that p_j is above p_k. Then the top-strip of p_i, denoted by $\overline{p_i}$, becomes the following plane subset:

$$\overline{p_i} = \{(x,y) \in \mathbf{R}^2 : y_i < y \leq y_k \quad \text{and} \quad x_j \leq x\}$$

If p_k is above p_j then we define $\overline{p_i}$ as

$$\overline{p_i} = \{(x,y) \in \mathbf{R}^2 : y_i < y \leq y_j \quad \text{and} \quad x \leq x_k\}$$

Analogously, the *bottom-, left-* or *right-strip* of a point could be defined. Later, the role of these strips will be clear.

Theorem 6. *Given a point set S, the following conditions are equivalent:*

(a) M_n is planar.
(b) No empty quadrilateral exists in S.
(c) The interior of every top-strip is empty.
(d) The interior of every top-, bottom-, left-, or right-strip is empty.

Proof. We prove that condition (b) is equivalent to the other three assertions. First, suppose that M_n is not geometrically planar; thus, the straight-line representation of M_n has two crossing edges. The extremes of these edges define an isothetic rectangle, which does not contain any other point in its interior, and so it is an empty quadrilateral. Reciprocally, consider an empty quadrilateral defined by four points. Then, the two pairs of opposite vertices must be adjacent in M_n and their edges, as straight-lines, have to intersect at a crossing point.

Let us prove now the equivalence between (b) and (c). Suppose that p_i, p_j, p_k, and p_l are four points in S forming an empty quadrilateral. Assume, without loss of generality, that p_i, p_j, p_k, and p_l are strictly higher, leftmost, rightmost, and lower, respectively, than the other three points, and that p_l is to the right of p_i. Considering all the possible cases (which appear in Fig. 7), we claim that the top-strip $\overline{p_i}$ is not empty. Firstly, if p_j is the point with closest abscissa to p_i to its left and p_l to its right, then $\overline{p_i}$ contains p_k. On the contrary, if any other point p is below p_i with closest abscissa to its left or right then p is necessarily below p_l and the top-strip $\overline{p_i}$ contains p_j or p_k. Finally, if two points p and p' are the points of closest abscissa at the left and right of p_i (again below it), then they lie beneath p_l and the top-strip contains p_j or p_k depending on whether p is below p' or vice versa. Conversely, if the top-strip $\overline{p_i}$ defined by p_i, p_j, and p_k contains p_l, then by sectioning the strip by an isothetic line through p_l we get our empty quadrilateral.

Similarly, the equivalence between (b) and (d) can be proved, and this completes the proof. ∎

This result gives rise to an algorithm for testing the planarity of M_n, which relies on assertion (c) of the theorem. This algorithm is divided into two steps: first, an auxiliary graph is constructed as the data structure, and second, the graph is transversed checking some conditions to obtain the final answer. To build the graph, the list of points is transversed twice.

ALGORITHM BUILD_G
let G be the graph having S as vertices and no edges and let L be the points $\{p_1, \ldots, p_n\}$ ordered from top to bottom;
for $j := 1$ **to** n **do**
  for every p_i with $i < j$ such that p_i is to the left of p_j, add the edge p_ip_j to G and delete p_i from L;
let L be $\{p_1, \ldots, p_n\}$;
for $j := 1$ **to** n **do**
  for every p_i with $i < j$ such that p_i lies to the right of p_j, add the edge p_ip_j to G and delete p_i from L;
return G.

Algorithm BUILD_G assures us that every point in G is joined to the two points below it to its left and its right with closest abscissa. This is all we need to test whether or not every top-strip $\overline{p_i}$ is empty, and according to Theorem 6, to conclude whether $M_n(S)$ is planar.

```
ALGORITHM PLANAR TEST DFG
let {p_1, ..., p_n} be the points labeled in the order from top to bottom;
let L be a list of them from left to right;
let G the graph obtained by running BUILD_G with {p_1, ..., p_n} as input;
for i := 1 to n do
    Search for the point p_i in L and let p_j and p_k be the previous
    and the following ones;
    m := max(j, k);
    if p_i p_m is not an edge of G then
        M_n is not planar and exit
    else
        delete p_i from L;
conclude that M_n is planar.
```

Proposition 7. *Algorithm PLANAR TEST DFG determines if there exists a planar dilation-free graph on a given point set. This is done in optimal $\Theta(n)$ time.*

Proof. Suppose p_i, p_j and p_k are three points satisfying the conditions of the first instruction inside the loop, and assume that $k > j$. If p_ip_k is not an edge of G, then a fourth point, namely p_l, is joined to p_i and is contained in the top-strip $\overline{p_i}$, as shown in Figure 8.

On the other hand, building the auxiliary graph is the step that dominates the computation. During this step, every vertex is considered at most once, and hence, the complete running time of the algorithm is $\Theta(n)$, which is the best possible. ∎

Once we test whether $M_n(S)$ of a given point set S is planar or not, let us see how $M_n(S)$ can be constructed. We propose a simpler though not faster algorithm than DFG consisting of transversing the points twice, from top to bottom and from bottom to top. Going down, every encountered point p_i is joined to the two points above it of closest abscissa to its left and right; and in the backward transversal every point is joined to those of closest abscissa to the left and the right, but beneath it.

```
ALGORITHM PLANAR DFG
let P := {p_1, p_2, ..., p_n} be the points ordered from top to bottom;
set L := {−∞, +∞};
set M_n to be the graph with P as vertex set and no edges;
for i := 1 to n do
    insert p_i in L;
    let r and s be the points next to p_i in L to its left and right;
    if r ≠ −∞ then
        add p_i r to M_n;
    if s ≠ +∞ then
        add p_i s to M_n;
set now P := {p_1, p_2, ..., p_n} as the points ordered from bottom to top;
set again L := {−∞, +∞};
for i := 1 to n do
    insert p_i in L;
    let r and s be the points next to p_i in L to its left and right;
    if r ≠ −∞ then
        add p_i r to M_n;
    if s ≠ +∞ then
        add p_i s to M_n;
return M_n as the output.
```

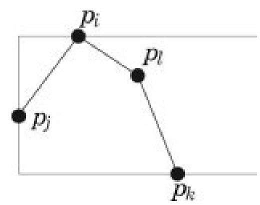

FIG. 8. A geometric proof of the algorithm's validity.

Theorem 8. *Algorithm PLANAR DFG yields the dilation-free graph M_n of a point set in optimal time $\Theta(n \log n)$.*

Proof. Suppose p_ip_j is an edge in M_n and assume that p_i is to the left of p_j and above it. If these two points were not joined along the top to bottom transversal when we meet p_i, then there exists a point p_k with closer abscissa than p_j to the right of p_i. Because p_ip_j is in M_n, their enclosing rectangle does not contain p_k and so p_k is lower than p_j. A similar reasoning can be argued when meeting p_j in the second transversal, and so a point p_l exists with closer abscissa than p_i at the left of p_j, and again, it should be placed above p_i. Thus, the interior of the top-strip $\overline{p_l}$ necessarily contains p_k, but this contradicts that M_n is planar as previously proved in Theorem 6.

It is obvious that the whole algorithm works in time $O(n \log n)$ and what only remains to be shown is that it is optimal. The basic idea of the proof is to project the points of S onto the bisector $y = x$, and thus to reduce it to an instance of the problem of sorting n numbers, which has a well-known lower bound of $O(n \log n)$. ∎

4. CONCLUSIONS

Although metrically speaking, dilation-free graphs in the Manhattan distance record the same information as complete graphs, we have proved that the former contains many fewer edges than the latter. We give one algorithm for constructing such a graph, running in $\Theta(n \log n + m)$ time, and a second one that tests in $O(\log n)$ time if a pair of points are joined in M_n after a preprocessing stage that works in $O(n \log n)$ time. In addition, we have also characterized whether the dilation-free graph of a set of points is planar or not, providing an algorithm for checking this condition that runs in $O(n \log n)$ time. Finally, whenever possible, this planar graph is algorithmically constructed in $O(n \log n)$ time.

Acknowledgments

The authors thank a referee of a previous version of the article for his/her valuable suggestions.

REFERENCES

[1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, Computational geometry: Algorithms and applications, Springer-Verlag, Berlin, 1997.

[2] L.P. Chew, There are planar graphs almost as good as the complete graph, J Comput System Sci 39 (1989), 205–219.

[3] R. Diestel, Graph theory, Springer-Verlag, New York, 1997.

[4] D. Eppstein, Dilation-free planar graphs, avalaible at the URL address http://www.ics.uci.edu/∼eppstein/junkyard/dilation-free/.

[5] D. Eppstein, "Spanning trees and spanners," Handbook of computational geometry, J.R. Sack and J. Urrutia (Editors), Elsevier Science, Amsterdam, 1999, pp. 425–461.

[6] J.M. Keil, Approximating the complete Euclidean graph, Proc. 1st. Scand. Workshop on Algorithm Theory, Springer LNCS 318, Springer, London, 1988, pp. 208–213.

[7] J.M. Keil and C.A. Gutwin, The Delaunay triangulation closely approximates the complete Euclidean graph, Discrete Comput Geom 7(1) (1992), 13–28.

[8] C. Levcopoulos and A. Lingas, There are planar graphs almost as good as the complete graph and as short as minimum spanning trees, Proc. Int. Symp. Optimal Algorithms, Springer LNCS 401, Springer, New York, NY, 1989, pp. 9–13.

[9] A. Okabe, B. Boots, and K. Sugihara, Spatial tessellations, John Wiley & Sons, Chichester, 1992.

[10] A.C. Yao, On constructing minimum spanning trees in k-dimensional space and related problems, SIAM J Comp 11 (1982), 721–736.