An Optimal Algorithm for the Rectilinear Link Center of a Rectilinear Polygon*

Bengt J. Nilsson† Sven Schuierer†

Juli 1991

Abstract

The problem of finding the link center of a simple polygon has been studied extensively in recent years. $O(n \log n)$ time upper bounds have been given for this problem and that of computing the link diameter for the polygon.

We consider the rectilinear case of this problem and give linear time algorithms to compute the rectilinear link diameter and the rectilinear link center of a simple rectilinear polygon. As a consequence we also obtain a linear time solution for the rectilinear link radius problem. To our knowledge these are the first optimal algorithms for center and diameter problems of polygons.

*This work was supported by the Deutsche Forschungs Gemeinschaft under Grant No. Ot 64/5–4.
†Institut für Informatik, Universität Freiburg, Rheinstr. 10–12, D-7800 Freiburg, Fed. Rep. of Germany.

1 Introduction

In many versions of motion planning problems the large cost involves making turns. An example is broadcasting a radio signal through a beam. At particular points relay stations must be erected to reflect the beam in a new direction. This leads to a cost measure for paths which is commonly known as the *link metric*.

The problem of finding link optimal paths between points inside a simple polygon has been solved by Suri [Sur87]. He gives a linear time algorithm for this problem. In addition he shows that the *link diameter*, the furthest link distance between any two points in a polygon, can be computed in $O(n \log n)$ time and linear storage (n will in the following always denote the number of polygon edges). A similar result is obtained by Ke [Ke89].

The *link center* problem is to find the set of points in a simple polygon from which the link distance to the furthest neighbour is minimum. It was first studied by Lenhart *et al.* [LPS*87]. They give a quadratic time algorithm for this problem which was later improved to $O(n \log n)$ by Djidjev *et al.* [DLS89] and Ke [Ke89].

In many cases we may want to restrict the paths to be rectilinear, i.e., all segments of the path are axis parallel. This particular restriction has been studied for simple rectilinear polygons by de Berg [B89]. He devises a data structure which allows to compute, given two query points inside the polygon, the shortest path between the points in $O(\log n + l)$ time, where l is the number of links of the path. The data structure can be constructed in $O(n \log n)$ time and requires $O(n \log n)$ storage. Furthermore, he shows how to compute the rectilinear link diameter in $O(n \log n)$ time with a simple divide and conquer approach.

In the more general case when the obstacles do not form the boundary of a simple polygon but instead are just rectilinear line segments in the plane, a generalization of the shortest path problem has been studied by de Berg *et al.* [BKNO90].

In this paper we give linear time algorithms for computing the rectilinear link diameter and the rectilinear link center of a simple rectilinear polygon. To our knowledge these are the first optimal algorithms found for center and diameter problems for non-trivial classes of polygons.

One should note the (conceptual) similarity between center finding algorithms in geometry and median finding algorithms for numeric lists. The link center can in some sense be viewed as the two dimensional median of the polygon and, hence, it is not astonishing that there exists a linear time algorithm for the rectilinear link center problem.

The basic idea to solve the link center problem is as follows. First, we construct a chord c which splits the polygon into two parts such that all the points of the link center are "close" to c. For our purposes close means that any point in the link center is at most two links away from the chord. The chord can be constructed by computing a diametral pair of vertices. The middle link of the shortest path between the two vertices can be extended until the end points touch the boundary of the polygon and the chord obtained in this way is a good candidate for c although it may have to be altered in order to conform with our requirements.

The second step of the algorithm consists of subdividing the region of points that can reach the chord with at most two links into a partial cover of histograms. This cover will consist of either one histogram, three histograms, or of one histogram plus a (possibly linear) number of histograms that are independent in a way which will be described later.

The last part of the algorithm is to apply a subroutine to each of the histograms found. The subroutine computes the part of the link center contained in the histogram in linear time.

The paper is organized as follows. In the next section we state our definitions and give some preliminary results for simple rectilinear polygons. The link center algorithm we develop needs to know both the diameter of the polygon and two vertices at distance equal to the diameter from each other. We show in Section 3 how to compute both of these with a linear time algorithm. Before we give the actual algorithm for the link center we present a necessary subroutine in Section 4 which shows how to compute the link center in linear time for a special case, i.e., the part of the link center contained in a maximal histogram having a rectilinear chord in the polygon as base. In Section 5 we present the algorithm to compute the link center, given a chord c such that there are points with great distance to c on both sides of c. Since the algorithm we state needs a chord having certain properties, Section 6 deals with the problem of computing the appropriate chord. Finally, in Section 7 we prove that the link center consists of at most two components.

2 Definitions

Let P be a Jordan curve consisting of n axis parallel line segments such that no two consecutive segments are collinear. We define a *simple rectilinear polygon* $\mathbf{P}$ to be the union of P and the interior of P. A (*rectilinear*) *path* $\mathcal{P}$ is a curve that consists of (a finite number of) axis parallel line segments inside $\mathbf{P}$. The length of $\mathcal{P}$, denoted by $\lambda(\mathcal{P})$, is the number of line segments it consists of. From now on we will only consider rectilinear polygons and rectilinear paths in a polygon $\mathbf{P}$. Hence, whenever we talk of polygons, we mean simple rectilinear polygons and whenever we talk of paths, we mean rectilinear paths in $\mathbf{P}$.

Let p and q be two points in a polygon $\mathbf{P}$ and let e be an axis parallel line segment in $\mathbf{P}$. The (*rectilinear*) *link distance* between p and q, denoted by $d(p,q)$, is defined as the length of the shortest path connecting p and q. We say a polygonal path $\mathcal{P}$ from p to line segment e is *admissible* if it is rectilinear and the last link of $\mathcal{P}$ is orthogonal to e. We define the link distance of p and e, again denoted by $d(p,e)$, to be the length of the shortest admissible path from p to a point of e.

The (*rectilinear*) *link diameter* denoted $D(\mathbf{P})$ is defined as the maximum rectilinear link distance between any two points in $\mathbf{P}$. We let $R(\mathbf{P})$ denote the (*rectilinear*) *link radius* which we define as the minimum integer k for which there is some point in $\mathbf{P}$ from which all other points can be reached with at most k links. When the polygon $\mathbf{P}$ is given we will refer to the radius as R. Finally, we define the (*rectilinear*) *link center* $lc(\mathbf{P})$ as the set of

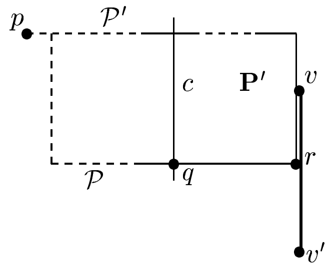

Figure 1: *One of v and v' is at least as far from p as q is.*

points in $\mathbf{P}$ that can reach all other points using at most R links.

In the rest of the paper the letter v with various subscripts and superscripts will always denote vertices of $\mathbf{P}$. In the same way the letters p, q, and r denote points of $\mathbf{P}$.

In the following lemma it is proven that there is always a vertex which is a furthest neighbour of a point of the polygon. This fact is used to discretize the problem, i.e., our link center algorithm will compute the set of points that are at most R links from each vertex.

LEMMA 2.1 *For any two points $p, q \in \mathbf{P}$, there is a vertex v such that $d(p,q) \leq d(p,v)$.*

PROOF: Let $k = d(p,q)$ and let $\mathcal{P}$ be a shortest path from p to q. Let r be the point of intersection between the boundary of $\mathbf{P}$ and the extension of the last link l of $\mathcal{P}$. Let $\mathcal{P}'$ be a shortest path from p to r. We will show that $\lambda(\mathcal{P}') \geq k$ using a proof by contradiction. So assume that $\lambda(\mathcal{P}') < k$. Obviously, q can not be a point on $\mathcal{P}'$. If the last link of $\mathcal{P}'$ is collinear with l, q can be reached from p with less than k links, a contradiction. On the other hand, if l is orthogonal to the last link of $\mathcal{P}'$, then follow the two paths $\mathcal{P}$ and $\mathcal{P}'$ from r until they join again. The path traced like this bounds a simple polygon $\mathbf{P}'$ interior to $\mathbf{P}$. The point q is obviously a point on the boundary of $\mathbf{P}'$. Also, since $\mathcal{P}$ is a shortest path, the chord c of $\mathbf{P}'$ through q and orthogonal to l has one end point on the path $\mathcal{P}'$ and, hence, by following the path $\mathcal{P}'$ to the intersection point with c and by continuing along the chord, a shorter path from p to q has been constructed. This proves that $d(p,r) = \lambda(\mathcal{P}') \geq k$; see Figure 1.

The point r lies on a boundary edge of $\mathbf{P}$ with vertices v and v'. Suppose both v and v' have distance less than k from p. Then, so does r since the line segment through r orthogonal to the edge vv' will join one of the paths from p to v or from p to v', a contradiction. Hence, either $d(p,v) \geq k$ or $d(p,v') \geq k$. □

This means that the diameter, radius, and link center definitions can be equivalently stated in terms of the vertices (v and v') of $\mathbf{P}$.

$$D(\mathbf{P}) \quad = \quad \max_{v,v' \in \mathbf{P}} d(v,v'),$$

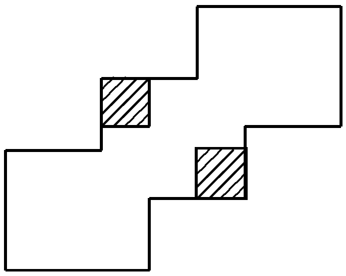

Figure 2: *The link center of a polygon can be disconnected.*

$$
\begin{aligned}
R(\mathbf{P}) &= \min_{p\in\mathbf{P}} \max_{v\in\mathbf{P}} d(p,v),\\
lc(\mathbf{P}) &= \{p \in \mathbf{P} \mid \text{for all} \quad v \in \mathbf{P}:\; d(p,v) \leq R(\mathbf{P})\}.
\end{aligned}
$$

The next lemma establishes upper and lower bounds on the radius of a polygon.

LEMMA 2.2

$$\lceil D(\mathbf{P})/2\rceil \leq R(\mathbf{P}) \leq \lceil D(\mathbf{P})/2\rceil + 1$$

PROOF: The first inequality is proven as follows. Take any two points $p, q \in \mathbf{P}$ and a point $r \in lc(\mathbf{P})$. This means that there is a path from p to r having at most $R(\mathbf{P})$ links and similarly a path from q to r of $R(\mathbf{P})$ links. Hence there is a path (the one via r) from p to q having at most $2R(\mathbf{P})$ links and therefore $D(\mathbf{P}) \leq 2R(\mathbf{P})$. The second inequality is proven in Section 6. □

The link diameter can be found in linear time; see Section 3. Lemma 2.2 implies that our linear time algorithm (which assumes a priori knowledge of the radius) for constructing the link center can be applied with only a slight modification, since we do not know the radius in advance. In order to do this, we execute the algorithm first with the value $R(\mathbf{P}) = \lceil D(\mathbf{P})/2\rceil$. If the algorithm produces a non-empty region, then we are done. Otherwise, $R(\mathbf{P}) = \lceil D(\mathbf{P})/2\rceil + 1$ and we run the algorithm with this value. This means that we can always assume that the algorithm knows the correct value of $R(\mathbf{P})$.

An interesting fact is that the rectilinear link center may be disconnected; see Figure 2. In the example the diameter is 4 and the radius is 2.

A *histogram* is a monotone rectilinear polygon with one of the monotone chains forming a single straight line segment. The single straight line segment forming one of the monotone chains is called the *base* of the histogram.

DEFINITION 2.1 A *maximal histogram* $\mathbf{H}$ inside a rectilinear polygon $\mathbf{P}$ having an axis parallel chord c in $\mathbf{P}$ as its base is the maximum area histogram interior to $\mathbf{P}$ with c as its base; see Figure 3.

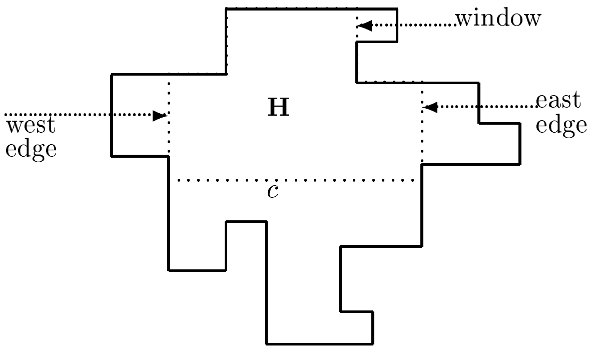

Figure 3: **H** *is a maximal histogram of the polygon.*

The definition does not yield a unique maximal histogram since **H** can extend to either side of c. However, if we specify a direction orthogonal to c the ambiguity can be resolved.

A *window* is a maximal segment of the boundary of a histogram which is not also part of the boundary of **P**. When the base is horizontal and all interior points of **H** lie above the base, we will use the term *west edge* to denote a vertical edge of **H** that has the interior to the right of the edge. An *east edge* is defined similarly as a vertical edge having the interior to the left.

We will make heavy use of the following construction. For an axis parallel line segment e and a point p in **P**, let $e(p, d)$ denote the part of e that can be reached from p with an admissible path of length at most d. de Berg shows in [B89] that, for any edge e of **P**, the set of distances $\{d(v, e) \mid v \in \mathbf{P}\}$ and the sets of subedges $\{e(v, d(v, e)) \mid v \in \mathbf{P}\}$ and $\{e(v, d(v, e) + 1) \mid v \in \mathbf{P}\}$ can be computed in linear time. It is important to note that $e(p, d_p)$ is a connected set, for any point p and any integer d_p. Because of this property, we will often refer to the subedges as *intervals*. Note that, in particular, the sets of subedges have the property that their intersection is non-empty if and only if the pairwise intersection of any two subedges is non-empty.

Since we rely heavily on these results we give informal arguments as to the truth of these statements. First, note that given the line segment e it is possible to partition the polygon into histograms such that e is an interval of the base of two such histograms; see [Lev87] for the algorithm. The partitioning can be computed in linear time given the vertex-edge pairs of [Cha90].

The path from a vertex to e will pass through the bases of all the histograms on the way and will make one turn in each histogram (except in the first). Thus, the distance $d(v, e)$ is easy to find using the dual graph of the histogram partition. Refer to Figure 4 for an example.

Moreover, de Berg shows in [B89] that for any vertex v with $d(v, e) > 2$, there is a vertex v_{next} such that the distance from v_{next} to e is one less than $d(v, e)$ and such that v and v_{next} have the same interval on e. The reason for this is that the second link of of a

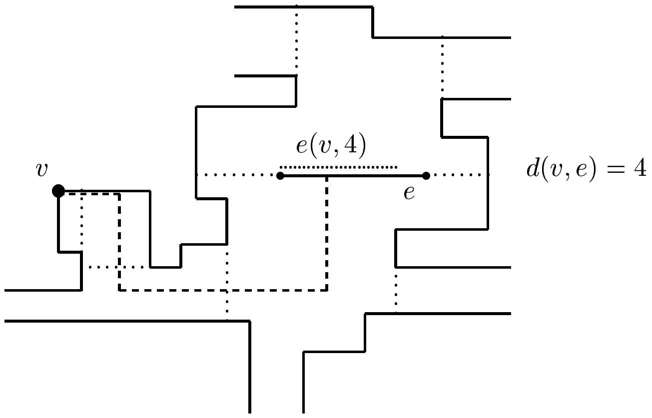

Figure 4: *A path from v to e of four links.*

shortest path between v and e can always be chosen so that it touches the boundary of $\mathbf{P}$. In other words, this means that the set of intervals $\{e(v, d(v, e)) \mid v \in \mathbf{P}\}$ only depends on the vertices that are exactly two links away from e and these can be computed in linear time. By keeping a reference at each vertex v to v_{next} we can obtain the intervals for all the vertices. For the set of intervals $\{e(v, d(v, e) + 1) \mid v \in \mathbf{P}\}$ a similar argument applies.

To see that the subedges are intervals it is enough to consider the histogram having the chord through e as base. A path from a point outside the histogram to e enters the histogram through a window and can then make a turn to reach the base. This turning point must lie within the interval between the window and the "opposite" side of the histogram.

The following lemma is also proven in [B89].

LEMMA 2.3 (DE BERG) *Let c be a chord which cuts $\mathbf{P}$ into two subpolygons such that p and q are points in different subpolygons and let $d(p, c) = d_p$ and $d(q, c) = d_q$. We have that*

$$\begin{array}{rcll} d(p,q) & = & d_p + d_q - 1 & \text{when } c(p, d_p) \cap c(q, d_q) \neq \emptyset, \\ & \geq & d_p + d_q & \text{otherwise.} \end{array}$$

This lemma will prove useful in later sections which is why we restate it here.

3 The Rectilinear Link Diameter

The first problem we consider is to how to compute the link diameter of a polygon $\mathbf{P}$. Recall that the diameter of $\mathbf{P}$ is the greatest distance between any two vertices in $\mathbf{P}$ and that we denote it by $D(\mathbf{P})$.

3.1 An Algorithm to Compute the Link Diameter

Our starting point is the following divide-and-conquer algorithm by de Berg [B89]. It makes use of the following fact. If we are given a chord c that splits $\mathbf{P}$ into two parts $\mathbf{P}_1$ and $\mathbf{P}_2$, then we have that the diameter is spanned either by two vertices in $\mathbf{P}_1$ or two vertices in $\mathbf{P}_2$ or one vertex in $\mathbf{P}_1$ and the other in $\mathbf{P}_2$. In the latter case the shortest path between the two vertices has to cross c. Hence, if we are given the intervals that the vertices in $\mathbf{P}_1$ and $\mathbf{P}_2$ can reach on c, then we can compute the maximal distance from a vertex in $\mathbf{P}_1$ to a vertex in $\mathbf{P}_2$ by Lemma 2.3.

(i) If $\mathbf{P}$ is a rectangle, then $D(\mathbf{P}) = 2$ and stop.

(ii) Compute a chord c of $\mathbf{P}$ that cuts $\mathbf{P}$ into two subpolygons $\mathbf{P}_1$ and $\mathbf{P}_2$ such that $|\mathbf{P}_1|$, $|\mathbf{P}_2| \leq 3/4\, n + 2$.

(iii) Compute $M = \max\{d(v, w) \mid v \in \mathbf{P}_1, w \in \mathbf{P}_2\}$ and remember the vertices v, w with $d(v, w) = M$.

(iv) Compute $D(\mathbf{P}_1)$ and $D(\mathbf{P}_2)$ recursively.

(v) Let $D(\mathbf{P}) := \max(M, D(\mathbf{P}_1), D(\mathbf{P}_2))$.

de Berg shows how Steps (ii) and (iii) can be carried out in linear time. This immediately leads to an $O(n \log n)$ algorithm. Our algorithm differs from the above one only in Step (iv). It is based on the following observation. For $i = 1, 2$, let d_i be the maximal distance from a vertex in $\mathbf{P}_i$ to c, i.e., $d_i = \max_{v \in \mathbf{P}_i} d(v, c)$. In the following discussion we assume that $d_1 \geq d_2$. There is one vertex v_i in each subpolygon $\mathbf{P}_i$ with $d(v_i, c) = d_i$, for $i = 1, 2$. For these we have $d(v_1, v_2) \geq d_1 + d_2 - 1$ again by Lemma 2.3. This gives us a lower bound of $d_1 + d_2 - 1$ on M. On the other hand, there is a trivial upper bound on the diameter of $\mathbf{P}_2$ which is $2d_2 + 1$ since all points of $\mathbf{P}_2$ can reach a point on c with d_2 links. Hence, if we can show that $D(\mathbf{P}_2) \leq d_1 + d_2 - 1$, we can avoid to recur on $\mathbf{P}_2$. Unfortunately, we cannot guarantee that $D(\mathbf{P}_2)$ is smaller than M. But we know that the interesting values of $D(\mathbf{P}_2)$ are in the range from $d_1 + d_2 - 1$ to $2d_2 + 1$. In the next section we will show that we can decide in linear time (in the size of $\mathbf{P}_2$) if $D(\mathbf{P}_2) \leq 2d_2 - 1 \leq M$ and, furthermore, if $D(\mathbf{P}_2) > 2d_2 - 1$, the exact value of $D(\mathbf{P}_2)$ can also be computed in linear time. As an immediate consequence we only have to recur on $\mathbf{P}_1$ in Step (iv).

3.2 Computing Bounds on the Diameter of a Polygon

For the rest of this section suppose that we are given a polygon $\mathbf{P}$ and one edge e of it. We denote the maximal distance of a vertex to e by d_e, i.e., $d_e = \max_{v \in \mathbf{P}} d(v, e)$. The problem we solve in this section is to show how to decide whether $D(\mathbf{P}) \leq 2d_e - 1$ and, if not, how to compute the exact value of $D(\mathbf{P})$, both in linear time. To this end we need the two sets of intervals $\mathcal{I}_1 = \{e(v, d_e) \mid v \in \mathbf{P}\}$ and $\mathcal{I}_2 = \{e(v, d_e + 1) \mid v \in \mathbf{P}\}$ both of which can be computed in linear time [B89, p. 10/11] given a triangulation which also can be computed in linear time by the algorithm of Chazelle [Cha90].

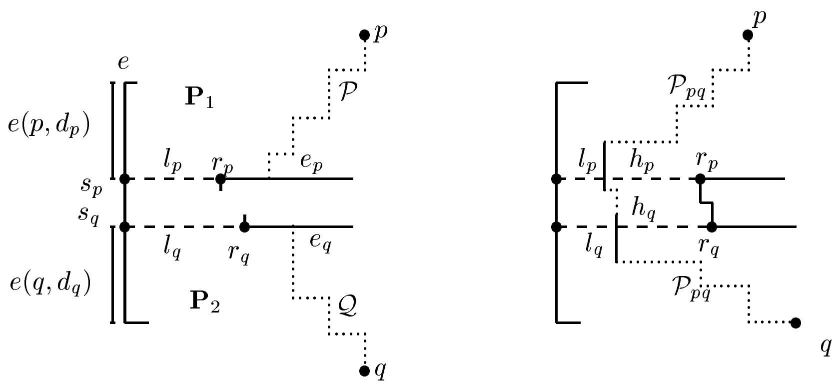

Figure 5: *A path between p and q has at least $d_p + d_q - 1$ links.*

The idea of the algorithm is to make use of the information that is supplied by the sets $\mathcal{I}_1$ and $\mathcal{I}_2$. In order to do so we make a few observations which are stated in the following three lemmas. The first lemma gives an upper bound on the distance between two points if the intervals on e that the points can reach with a given number of links overlap.

LEMMA 3.1 *If p and q are two points in $\mathbf{P}$ and d_p and d_q are two integers such that $e(p, d_p) \cap e(q, d_q) \neq \emptyset$, then $d(p, q) \leq d_p + d_q - 1$.*

PROOF: Let s be a point in $e(p, d_p) \cap e(q, d_q)$. Consider two admissible paths $\mathcal{P}$ from p to s and $\mathcal{Q}$ from q to s with d_p and d_q links, respectively. Since the last link is orthogonal to e in both cases, we can join these two paths into one by leaving aside the shorter of the two last links. The new path obviously has length $d_p + d_q - 1$. Hence, $d(p, q) \leq d_p + d_q - 1$. □

An immediate consequence of Lemma 3.1 is that if $\bigcap \mathcal{I}_1 \neq \emptyset$,[1] then $D(\mathbf{P}) \leq 2d_e - 1 \leq M$ since all the intervals $e(v, d_e)$ with $v \in \mathbf{P}$ overlap. The second lemma gives us a lower bound on the distance between two points if their intervals do not overlap.

LEMMA 3.2 *If p and q are points in $\mathbf{P}$ and d_p and d_q are two integers such that $e(p, d_p)$ and $e(q, d_q)$ are non-empty and $e(p, d_p) \cap e(q, d_q) = \emptyset$, then $d(p, q) \geq d_p + d_q - 1$.*

PROOF: W.l.o.g. assume that e is vertical, that the interior of $\mathbf{P}$ is to the right of e, and that $e(p, d_p)$ is above $e(q, d_q)$. Let s_p be the lower end point of $e(p, d_p)$ and s_q be the upper end point of $e(q, d_q)$. Further, let $\mathcal{P}$ be an admissible path of length d_p from p to s_p with last link l_p and $\mathcal{Q}$ be an admissible path of length d_q from q to s_q with last link l_q (see Figure 5).

Since we cannot move l_p lower resp. l_q higher, there is a horizontal upward (downward) edge of $\mathbf{P}$ that overlaps l_p (l_q). Since we allow more than two vertices to be collinear, l_p

[1]We denote the intersection of all sets in a family $\mathcal{F}$ by $\bigcap \mathcal{F}$.

(l_q) may overlap a number of horizontal edges. We choose e_p (e_q) to be the leftmost of these. Let r_p be the left end point of e_p and r_q be the left end point of e_q. Furthermore, let h_p be the horizontal line segment from r_p to e and h_q the horizontal line segment from r_q to e. We denote the part of $\mathbf{P}$ above[2] h_p by $\mathbf{P}_1$ and the part of $\mathbf{P}$ below h_q by $\mathbf{P}_2$.

Our first claim is that p is in $\mathbf{P}_1$. To see this assume the contrary. Since the right end point of l_p is contained in $\mathbf{P}_1$ and the boundary of $\mathbf{P}_1$ is a Jordan curve, $\mathcal{P}$ crosses h_p. Let l be the first vertical link that intersects h_p. Since there is no upward edge of $\mathbf{P}$ that overlaps h_p, we can connect a point of l slightly below h_p with one horizontal link to e which contradicts the fact that we chose s_p to be the lowest point on e that can be connected to p with d_p or less links.

Since we cannot reach any point below s_p from p with d_p links, $\mathcal{P}$ cannot cross h_p and, hence, p is in $\mathbf{P}_1$. For similar reasons q is in $\mathbf{P}_2$. Therefore, a shortest path $\mathcal{P}_{pq}$ from p to q has to cross both h_p and h_q. Let l_1 be the first vertical link from p that intersects h_p. The part of $\mathcal{P}_{pq}$ from to p to l_1 can have no less than $d_p - 1$ links since, otherwise, p can reach points below s_p with d_p links as we argued above. Let l_2 be the first vertical link from q that intersects h_q. For the same reason as above, the part of $\mathcal{P}_{pq}$ from q to l_2 has at least $d_q - 1$ links. Hence, the length of $\mathcal{P}_{pq}$ is at least $(d_p - 1) + (d_q - 1) + 1$. □

An immediate consequence of Lemma 3.2 is that if $\bigcap \mathcal{I}_2$ is empty, then $D(\mathbf{P}) = 2d_e + 1$. To see this first note that, for a family $\mathcal{I}$ of intervals, we have $\bigcap \mathcal{I} = \emptyset$ if and only if there are two intervals I_1, $I_2 \in \mathcal{I}$ with $I_1 \cap I_2 = \emptyset$. Hence, there are two vertices v and v' in $\mathbf{P}$ with $e(v, d_e + 1) \cap e(v', d_e + 1) = \emptyset$ and Lemma 3.2 yields that $d(v, v') \geq 2d_e + 1$. On the other hand, we noted in the previous section that $D(\mathbf{P}) \leq 2d_e + 1$ and, thus, we obtain that $D(\mathbf{P}) = 2d_e + 1$.

Unfortunately, it is not always true that either $\bigcap \mathcal{I}_1$ is non-empty or $\bigcap \mathcal{I}_2$ is empty. So from now on suppose that $\bigcap \mathcal{I}_1 = \emptyset$ and $\bigcap \mathcal{I}_2 \neq \emptyset$. What are the remaining cases? Consider two points p and q. Since $\bigcap \mathcal{I}_1 = \emptyset$, we can assume that $e(p, d_e) \cap e(q, d_e) = \emptyset$. Furthermore, we have that $e(p, d_e + 1) \cap e(q, d_e + 1) \neq \emptyset$ because $\bigcap \mathcal{I}_2$ is non-empty. Hence, we have to look at the intersections $e(p, d_e) \cap e(q, d_e + 1)$ and $e(p, d_e + 1) \cap e(q, d_e)$. Either both of them contain at least one point or at least one of the two intersections is empty. The former case (both intersections are non-empty) is dealt with in the following lemma. As we will show later the latter case can also be treated with the tools we have developed so far.

Lemma 3.3 *If p and q are two points in $\mathbf{P}$ and d_p and d_q are two integers such that*

(i) $e(p, d_p)$ and $e(q, d_q)$ are non-empty,

(ii) $e(p, d_p) \cap e(q, d_q) = \emptyset$,

(iii) $e(p, d_p + 1) \cap e(q, d_q) \neq \emptyset$, and

(iv) $e(p, d_p) \cap e(q, d_q + 1) \neq \emptyset$,

[2]If we speak of polygons, in this case $\mathbf{P}_1$ and $\mathbf{P}_2$, being *above* or *below* a line segment, we mean that the interior of the curve is locally above or below the line segment.

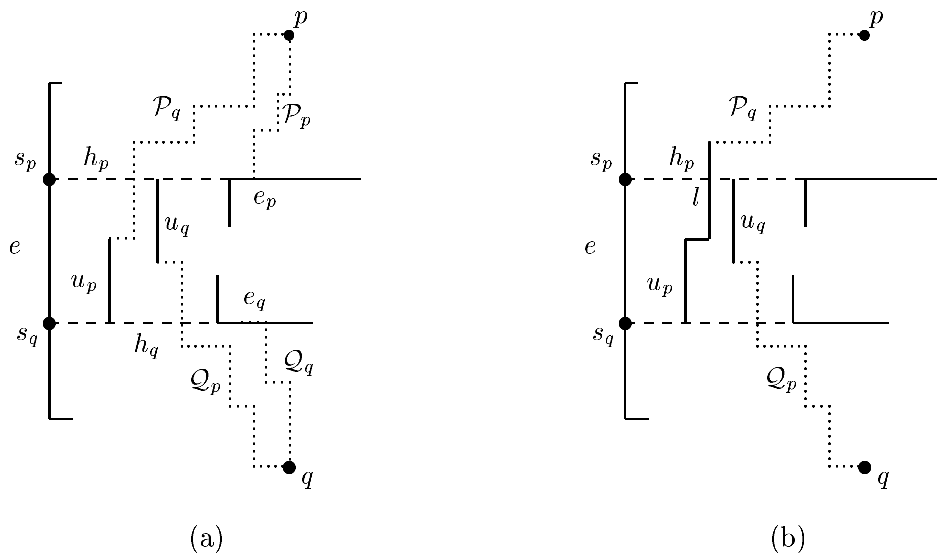

Figure 6: *There is a horizontal link between h_p and h_q.*

then $d(p,q) = d_p + d_q - 1$.

PROOF: By Property (ii) and Lemma 3.2 we have that $d(p,q) \geq d_p + d_q - 1$. Hence, we only have to show the reverse inequality. Before we start proving the claim we give a precise definition of the situation we consider. As before we assume that e is vertical, that the interior of the polygon is to the right of e, and that $e(p,d_p)$ is above $e(q,d_q)$. We again denote the lowest point of $e(p,d_p)$ by s_p and the highest point of $e(q,d_q)$ by s_q. Further, let $\mathcal{P}_p$ be an admissible path from p to s_p of length d_p. Since we cannot reach any point below s_p with at most d_p links, the last link of $\mathcal{P}_p$ overlaps with some horizontal edges. We denote the leftmost one of these by e_p. Let h_p be the horizontal line segment from e_p to s_p. Furthermore, let $\mathcal{Q}_q$, e_q and h_q be defined analogously for q. Let $\mathcal{P}_q$ be an admissible path from p to s_q (having length d_p+1) and $\mathcal{Q}_p$ be an admissible path from q to s_p (of length d_q+1). Note that these exist since $e(p,d_p+1) \cap e(q,d_q)$ is non-empty as is $e(p,d_p) \cap e(q,d_q+1)$. As we have argued in Lemma 3.2, $\mathcal{P}_q$ crosses h_p and $\mathcal{Q}_p$ crosses h_q. It is our aim to show that $\mathcal{P}_q$ and $\mathcal{Q}_p$ can be joined to yield a path of length $d_p + d_q - 1$ that connects p and q. Let u_p be the d_p^{th} link of $\mathcal{P}_q$ and u_q be the d_q^{th} link of $\mathcal{Q}_p$. Clearly, u_p and u_q are vertical. For illustration see Figure 6a.

The proof is now organized as follows. First we show that we can assume that the second to last links u_p and u_q of $\mathcal{P}_q$ and $\mathcal{Q}_p$ both span the part from h_p to h_q and that $\mathcal{P}_p$ and $\mathcal{Q}_q$ are shortest paths. In the second part we show how to construct a path of length $d_p + d_q - 1$ if the rectangle $\mathbf{R}$ spanned by the upper end point s_1 of u_p and the upper end point s_2 of u_q is empty, i.e., contains no points of the boundary of $\mathbf{P}$. Finally, we proof that the rectangle $\mathbf{R}'$ spanned by s_1 and the right end point r of the last link of $\mathcal{P}_p$ is empty.

Since it can be easily seen that $\mathbf{R}'$ contains $\mathbf{R}$, this completes the proof.

Part 1

First assume that one of the paths, say $\mathcal{P}_q$, has a horizontal link between h_p and h_q as shown in Figure 6b. Let l be the first vertical link of $\mathcal{P}_q$ that intersects h_p. Since there is a horizontal link of $\mathcal{P}_q$ between h_p and h_q, l is at least the fourth link from the end of $\mathcal{P}_q$. Hence, if we connect u_q with one horizontal link to l, the resulting path has length at most $\lambda(\mathcal{P}_q) - 3 + \lambda(\mathcal{Q}_p) - 1 + 1 = d_p - 2 + d_q + 1 = d_p + d_q - 1$.

With a similar argument we can show the claim if s_p can be reached from p with less than d_p links or s_q from q with less than d_q links. Suppose that this is the case, say we can reach s_p with an admissible path $\mathcal{P}'$ consisting of $d_p - 1$ links. Then, we just connect u_q to $\mathcal{P}'$ by extending the last (horizontal) link l of $\mathcal{P}'$ towards u_q. This is always possible since either l contains the last link of $\mathcal{P}_p$ and, hence, connects to u_q or l is contained in the last link of $\mathcal{P}_p$ and then the horizontal line segment from the end point of u_q to l does not cross the boundary of $\mathbf{P}$. This again yields a path of length $\lambda(\mathcal{P}') + \lambda(\mathcal{Q}_p) - 1 = d_p - 1 + d_q + 1 - 1 = d_p + d_q - 1$.

So we can assume that both u_p and u_q intersect h_p and h_q and that $\mathcal{P}_p$ and $\mathcal{Q}_q$ are shortest paths. Since we cannot reach s_q from p with d_p links, $\mathcal{P}_q$ is also a shortest path and so is $\mathcal{Q}_p$.

Part 2

Next we show how to construct a path of length $d_p + d_q - 1$. W.l.o.g. assume that u_p is to the left of u_q. Let s_1 be the upper end point of u_p and let s_2 be the upper end point of u_q. Note that by our assumption on u_p s_1 is above h_p. For now suppose that the horizontal line segment l_1 from s_1 to the x-coordinate of s_2 and the vertical line segment l_2 from s_2 to the y-coordinate of s_1 do not cross the boundary of $\mathbf{P}$. We can construct a path from p to q of length $d_p + d_q - 1$ in the following way. We take the part of $\mathcal{Q}_p$ from q to u_q and append l_2 to it which yields a path of length d_q. We now claim that we can connect the end point of l_2 to the $(d_p - 2)^{\text{nd}}$ link of $\mathcal{P}_q$ with one link. Once this is shown the resulting path has length $d_q + (d_p - 2) + 1 = d_p + d_q - 1$. Let h_1 be the horizontal link of $\mathcal{P}_q$ before u_p. Since l_1 intersects h_1 in s_1 and l_1 and h_1 are collinear, $l_1 \cup h_1$ is a line segment that contains the upper end point of l_2 and the end point of the vertical link u_1 which occurs before h_1 in $\mathcal{P}_q$. Hence, we can connect u_1 to l_2 with one link. Since u_p is the d_p^{th} link of $\mathcal{P}_q$, u_1 is the $(d_p - 2)^{\text{nd}}$ link (see Figure 7) and the resulting path has length $d_p - 2 + 1 + d_q = d_p + d_q - 1$.

Part 3

Hence, we only have to prove that $l_1 \cup l_2$ does not cross the boundary of $\mathbf{P}$. Let $\mathbf{R}$ be the rectangle spanned by s_1 and s_2. Clearly, the boundary of $\mathbf{P}$ does not cross $l_1 \cup l_2$ if $\mathbf{R}$ contains no boundary points of $\mathbf{P}$ in its interior.

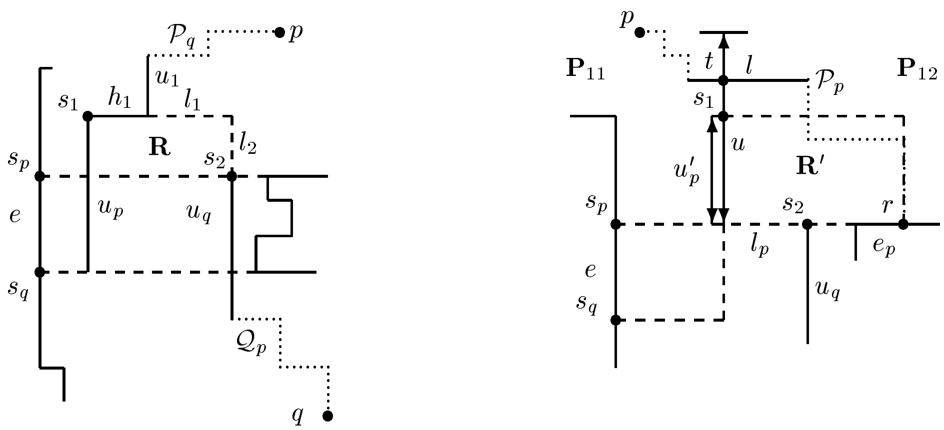

Figure 7: $\mathbf{R}$ *has to be empty.*

To see that the interior of $\mathbf{R}$ does not cross the boundary of $\mathbf{P}$ we proceed as follows. Let l_p be the last link of $\mathcal{P}_p$ and let u'_p be the part of u_p above h_p. Since s_2 is on l_p, $\mathbf{R}$ is contained in the rectangle $\mathbf{R}'$ spanned by the right end point r of l_p and s_1. Therefore, it suffices to show that there is no point of the boundary of $\mathbf{P}$ in $\mathbf{R}'$.

As we have argued in the proof of Lemma 3.2, p is in the part $\mathbf{P}_1$ of $\mathbf{P}$ that is above h_p. We extend u'_p to a vertical segment u that touches the boundary of $\mathbf{P}_1$ at both end points. u splits $\mathbf{P}_1$ into two parts, $\mathbf{P}_{11}$ to the left of u and $\mathbf{P}_{12}$ to the right of u. It is easy to see that p is in $\mathbf{P}_{12}$. In order to prove this suppose the contrary; then, $\mathcal{P}_p$ crosses u twice since r is in $\mathbf{P}_{12}$ and l_p also intersects u. The idea is now to take the first intersection point t of $\mathcal{P}_p$ with u and drop down along u. An easy calculation shows that the horizontal link l of $\mathcal{P}_p$ t belongs to is at most the $(d_p - 2)^{\text{nd}}$ link of $\mathcal{P}_p$ as seen from p. So we can make a shortcut when reaching u. We just drop immediately to l_p. Since u intersects l_p to the left of e_p we can even prolong u below l_p. But then we can reach points below s_p with d_p links. This contradicts the fact that s_p is the lower end point of $e(p, d_p)$. Hence, $\mathcal{P}_p$ does not cross u before l_p and p is in $\mathbf{P}_{12}$. Obviously, $\mathcal{P}_q$ does not cross u either since, if it does, we can shorten $\mathcal{P}_q$ by at least one link.

Now that we have shown that p is contained in $\mathbf{P}_{12}$ we turn to proving that no boundary point of $\mathbf{P}$ intersects the interior of $\mathbf{R}'$. The proof is again by contradiction. So suppose that there is a boundary point x of $\mathbf{P}$ in the interior of $\mathbf{R}'$ (see Figure 8). We will show that either $\mathcal{P}_p$ or $\mathcal{P}_q$ is not a shortest path in this case. We choose x to be one of the leftmost points of the boundary of $\mathbf{P}$ in $\mathbf{R}'$ (which may be on u'_p). We denote the horizontal line segment from x to u'_p by h_x (if x is on u'_p, then h_x degenerates to x). h_x partitions $\mathbf{P}_{12}$ into two parts, $\mathbf{P}_a$ above h_x and $\mathbf{P}_b$ below h_x. We distinguish two cases according to the location of p.

Case 1 p is in $\mathbf{P}_a$.
We show that we can shorten $\mathcal{P}_p$ in this case. $\mathcal{P}_p$ crosses h_x since r is in $\mathbf{P}_b$. Let l be

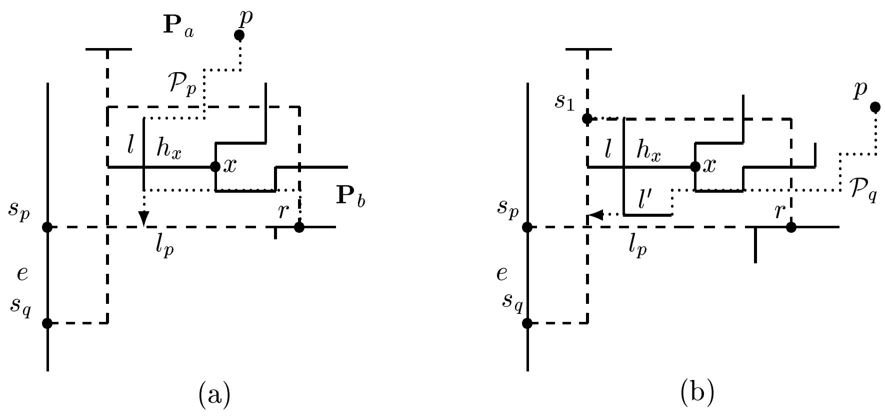

Figure 8: *There is a boundary point in* $\mathbf{R}'$.

the first vertical link of $\mathcal{P}_p$ that intersects h_x. Since x is one of the leftmost points of the boundary of $\mathbf{P}$ in $\mathbf{R}'$, we can prolong l down so that it reaches l_p which forms the lower boundary of $\mathbf{R}'$. On the other hand, we need at least one more link to reach r from l. This contradicts the fact that we assumed $\mathcal{P}_p$ to be a shortest path.

Case 2 p is in $\mathbf{P}_b$.
We show that then $\mathcal{P}_q$ can be shortened. $\mathcal{P}_q$ crosses h_x, since the upper end point s_1 of u_p belongs to $\mathbf{P}_a$. Let l be again the first vertical link of $\mathcal{P}_q$ that intersects h_x. x is in this case clearly to the right of u'_p since otherwise l would be collinear with u_p which immediately contradicts the fact that $\mathcal{P}_q$ is a shortest path.

Since at least one more link is needed to reach s_1 as the upper end point of the second to last link of $\mathcal{P}_q$ from l, l is at most the $(d_p - 2)^{\text{nd}}$ link of $\mathcal{P}_q$ as seen from p. This implies that l cannot intersect l_p which is the d_p^{th} link in $\mathcal{P}_p$. Otherwise, we can construct a path from p to s_p that is shorter than $\mathcal{P}_p$.

Hence, the horizontal link l' before l on $\mathcal{P}_q$ is *above* l_p, i.e., it is (partially) contained in $\mathbf{R}'$. Since there are no boundary points of $\mathbf{P}$ to the left of x and, therefore, to the left of l, we can extend l' to the left until it intersects u_p. The resulting path from p to s_q has at least two links less than $\mathcal{P}_q$ which contradicts the fact that $\mathcal{P}_q$ is a shortest path from p to s_q.

It may occur that l is the very first link of $\mathcal{P}_q$. This can be taken care of by the observation that, then, we can connect p with one horizontal link to u_p instead of using first a vertical and then a horizontal link.

Hence, the assumption that the boundary of $\mathbf{P}$ intersects the interior of $\mathbf{R}'$ leads to a contradiction in both cases. This, in particular, implies that the two line segments l_1 and l_2 that we need for the construction of a path of length $d_p + d_q - 1$ from p to q do not cross the boundary of $\mathbf{P}$. Thus, the construction of the path is possible and we can connect p and q with $d_p + d_q - 1$ links as claimed in the lemma. This completes the proof. □

With the help of the above three lemmas we are now in a position to attack the problem of computing the claimed bounds on $D(\mathbf{P})$. We already have considered the case that $\bigcap \mathcal{I}_1$ is non-empty and that $\bigcap \mathcal{I}_2$ is empty; we have shown that we obtain the desired bounds on $D(\mathbf{P})$ in these cases. So suppose that we have that $\bigcap \mathcal{I}_1 = \emptyset$ and at the same time that $\bigcap \mathcal{I}_2 \neq \emptyset$. In the following lemma we summarize the results and also deal with the two remaining cases that can occur.

LEMMA 3.4 *If $\mathbf{P}$ is a polygon, e is an edge of $\mathbf{P}$, and $d_e = \max_{v \in \mathbf{P}} d(v, e)$, then we have that*

(i) if $\bigcap \mathcal{I}_1 \neq \emptyset$, then $D(\mathbf{P}) \leq 2d_e - 1$;

(ii) if $\bigcap \mathcal{I}_2 = \emptyset$, then $D(\mathbf{P}) = 2d_e + 1$;

(iii) if $\bigcap \mathcal{I}_1 = \emptyset$ and $\bigcap \mathcal{I}_2 \neq \emptyset$, then

  (a) if, for all v_1, $v_2 \in \mathbf{P}$, we have that $e(v_1, d_e) \cap e(v_2, d_e+1) \neq \emptyset$, then $D(\mathbf{P}) = 2d_e - 1$;

  (b) if there are two points v_1, $v_2 \in \mathbf{P}$ with $e(v_1, d_e) \cap e(v_2, d_e + 1) = \emptyset$, then $D(\mathbf{P}) = 2d_e$.

PROOF: Claims (i) and (ii) have been dealt with previously. So we immediately turn to proving Claims (iii.a) and (iii.b). First consider Claim (iii.a). By Lemma 3.2 we have that $D(\mathbf{P}) \geq 2d_e - 1$ since $\bigcap \mathcal{I}_1 = \emptyset$ implies that there is a pair of vertices v' and v'' with $e(v', d_e) \cap e(v'', d_e) = \emptyset$. Now let v_1 and v_2 be two arbitrary vertices in $\mathbf{P}$. We have to show that we can connect them with a path of length at most $2d_e - 1$. If $e(v_1, d_e) \cap e(v_2, d_e) \neq \emptyset$, then Lemma 3.1 yields the desired result. If $e(v_1, d_e) \cap e(v_2, d_e) = \emptyset$, then we can apply Lemma 3.3 since by Condition (iii.a) we have that $e(v_1, d_e) \cap e(v_2, d_e + 1) \neq \emptyset$ and $e(v_1, d_e + 1) \cap e(v_2, d_e) \neq \emptyset$. This yields that there is a path of length $2d_e - 1$ that connects v_1 and v_2.

Now consider Claim (iii.b). Let v' and v'' be two vertices in $\mathbf{P}$ with $e(v', d_e) \cap e(v'', d_e + 1) = \emptyset$ which exist by assumption, then Lemma 3.2 yields that $d(v', v'') \geq 2d_e$ and, therefore, $D(\mathbf{P}) \geq 2d_e$. So now let v_1 and v_2 be two arbitrary vertices in $\mathbf{P}$. We want to show that $d(v_1, v_2) \leq 2d_e$. If $e(v_1, d_e) \cap e(v_2, d_e + 1) \neq \emptyset$, then we have by Lemma 3.1 that $d(v_1, v_2) \leq 2d_e$. Hence, assume that $e(v_1, d_e) \cap e(v_2, d_e + 1) = \emptyset$. Since $\bigcap \mathcal{I}_2 \neq \emptyset$, we have $e(v_1, d_e + 1) \cap e(v_2, d_e + 1) \neq \emptyset$, and, further, by the definition of d_e we have $e(v_1, d_e) \neq \emptyset$ and $e(v_2, d_e + 2) = e$ so that $e(v_1, d_e) \cap e(v_2, d_e + 2) \neq \emptyset$. Hence, the prerequisites of Lemma 3.3 are met if we consider $p = v_1$, $q = v_2$, $d_p = d_e$ and $d_q = d_e + 1$. Thus, we have $d(v_1, v_2) = d_p + d_q - 1 = d_e + d_e + 1 - 1 = 2d_e$. □

Given the two sets of intervals $\mathcal{I}_1$ and $\mathcal{I}_2$, we can easily compute their intersections and test whether $\bigcap \mathcal{I}_1 \neq \emptyset$ or $\bigcap \mathcal{I}_2 = \emptyset$. If none of the two conditions hold, we have to test for the cases of Lemma 3.4. Note that there are two vertices v_1 and v_2 such that $e(v_1, d_e) \cap e(v_2, d_e + 1) = \emptyset$ if and only if there exists a vertex v such that $e(v, d_e) \cap \bigcap \mathcal{I}_2 = \emptyset$. Hence, this test can also be carried out in time proportional to the number of intervals in $\mathcal{I}_1$ and $\mathcal{I}_2$ which yields an overall linear time algorithm.

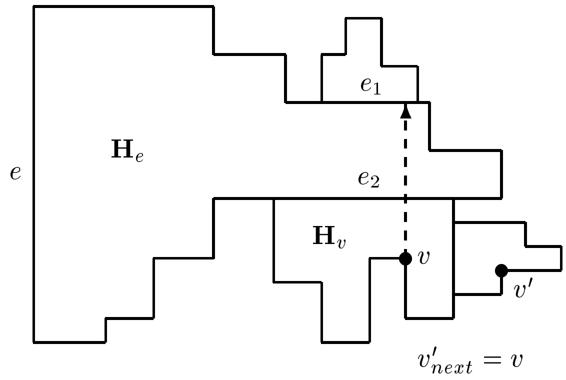

Figure 9: *How to obtain a sorted interval list.*

3.3 Computing a Sorted Interval List

In order to execute Step (iii) of the link diameter algorithm presented previously, de Berg not only needs to compute all the intervals $e(v, d(v, e))$, for edge e, but he also needs them sorted according to the upper and lower end points, for vertices at a given distance. In the following we show how to obtain a sorted list in linear time. W.l.o.g. assume that e is vertical. de Berg computes, for each vertex v at distance $d(v, e) > 2$, a vertex v_{next} that has a distance of one less to e and that has the same interval as v. This can be done in time linear in the number of vertices of $\mathbf{P}$. We now keep a list $\mathcal{L}_v$, at each vertex v in $\mathbf{P}$, of the vertices v' with $v'_{next} = v$. Since $\mathcal{L}_v$ is unsorted, building it costs a constant amount of time, for each vertex v' that is added to $\mathcal{L}_v$. Furthermore, for a vertex v that needs only two links to reach e, de Berg computes the edge e_1 of $\mathbf{H} \cup \mathbf{H}_v$ that can be reached from v by an axis parallel line segment where $\mathbf{H}$ is the maximal histogram induced by e (in $\mathbf{P}$) and $\mathbf{H}_v$ is the histogram containing v (see Figure 9). W.l.o.g. assume that e_1 is a north edge of $\mathbf{H}$. At this edge we keep a list of all the vertices (at distance two to e) that have the y-coordinate of e_1 as the upper end point of their interval. At the edge e_2 that is collinear with the base of $\mathbf{H}_v$, we keep a list of all the vertices that have the y-coordinate of e_2 as the lower end point of their interval.

In order to obtain a list of the intervals that is sorted according to the lower end points we proceed as follows. We traverse the histogram $\mathbf{H}$ from bottom to top. Since $\mathbf{H}$ is monotone w.r.t. the y-axis we encounter the horizontal edges in sorted order. At each horizontal edge e we follow the list $\mathcal{L}_e$ of the vertices at distance 2 whose intervals have the y-coordinate of e as the lower end point of their interval. For each vertex v in $\mathcal{L}_e$, we follow the vertices in the list $\mathcal{L}_v$ backward until the vertices at a given distance are found which are then added to the sorted list together with their intervals. To obtain a list of the intervals that is sorted according to upper end points we proceed analogously.

Since the computation of the intervals $e(v, d(v, e) + 1)$ is similar to the above described procedure we can also sort them in linear time if we keep the list $\mathcal{L}_v$ of backward references introduced above.

The algorithm to compute the link diameter of a polygon $\mathbf{P}$ can now be stated as follows.

(i) If $\mathbf{P}$ is a rectangle, then $D(\mathbf{P}) = 2$ and stop.
(ii) Compute a chord c of $\mathbf{P}$ that cuts $\mathbf{P}$ into two subpolygons $\mathbf{P}_1$ and $\mathbf{P}_2$ such that $|\mathbf{P}_1|$, $|\mathbf{P}_2| \leq 3/4\,n + 2$.
(iii) Compute $m = \max\{d(v, v') \mid v \in \mathbf{P}_1, v' \in \mathbf{P}_2\}$ and remember the vertices v, v' with $d(v, v') = m$.
(iv) Compute d_1 and d_2. If $d_1 < d_2$, then exchange $\mathbf{P}_1$
(v) Compute $M = \max(m, D(\mathbf{P}_2))$.
  (v.a) Compute $\mathcal{I}_1 = \{e(v, d_2) \mid v \in \mathbf{P}_2\}$ and $\mathcal{I}_2 = \{e(v, d_2 + 1) \mid v \in \mathbf{P}_2\}$.
  (v.b) **if** $\bigcap \mathcal{I}_2 = \emptyset$
    then $M := \max(m, 2d_2 + 1)$
    else if there is a vertex v in $\mathbf{P}_2$ with $e(v, d_2) \cap \bigcap \mathcal{I}_2 = \emptyset$
    then $M := \max(m, 2d_2)$
    else $M := m$;
    (* Note that we only have to check for the Cases (ii) and (iii.b) of Lemma 3.4 since in Cases (i) and (iii.a) we have that $D(\mathbf{P}_2) \leq 2d_2 - 1 \leq m$. *)
(vi) Compute $D(\mathbf{P}_1)$ recursively.
(vii) Let $D(\mathbf{P}) := \max(M, D(\mathbf{P}_1))$.

The function $T(n)$ for the number of steps which are needed by the above algorithm for an n-vertex polygon satisfies the following recurrence.

$$\begin{aligned} T(n) &\leq T(3/4\,n) + O(n) \\ T(1) &= O(1) \end{aligned}$$

which yields that $T(n) = O(n)$. Thus, we have proven the following theorem.

THEOREM 1 *The rectilinear link diameter of a simple rectilinear polygon can be computed in time linear in the number of vertices of the polygon.*

4 The Solution to a Special Case

In this section we will establish a solution to a special case of the link center problem. Suppose we are given a polygon $\mathbf{P}$, its radius R, and a maximal histogram $\mathbf{H}$ along an axis parallel chord of the polygon. We consider the question of how to output the part of the link center contained in $\mathbf{H}$.

Let c be a horizontal chord in $\mathbf{P}$ and $\mathbf{H}$ a horizontally monotone histogram with base c; see Definition 2.1. We assume that all interior points of $\mathbf{H}$ lie above c.

In the following we denote by $\mathbf{P}_w$ the subpolygon cut off by window w of $\mathbf{H}$.

Definition 4.1 The region $\mathbf{A}_w(d)$ associated with window w is the region in $\mathbf{H}$ which can reach all points in $\mathbf{P}_w$ using at most d links.

Next, we prove that the intersection of all the regions $\mathbf{A}_w(R)$ is exactly the part of the link center contained in $\mathbf{H}$.

Lemma 4.1 *Let $w_1, \ldots, w_m$ be the windows of $\mathbf{H}$.*

If $R = 2$, then $lc(\mathbf{P}) \cap \mathbf{H}$ is equal to the intersection $\bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R)$ intersected with $lc(\mathbf{H})$, the link center of $\mathbf{H}$.

If $R \ge 3$, then $lc(\mathbf{P}) \cap \mathbf{H}$ is equal to $\bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R)$.

Proof: Assume first that $R \ge 3$. This means that we can disregard the vertices of $\mathbf{H}$, since any such vertex can reach any point in $\mathbf{H}$ with at most three links. Hence, we only need to concern ourselves with the vertices contained in the subpolygons cut off by the windows of $\mathbf{H}$. Take a point $p \in \bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R)$. From p it is possible to reach the furthest vertex in each subpolygon $\mathbf{P}_{w_i}$ using at most R links. Hence, $\bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R) \subseteq lc(\mathbf{P}) \cap \mathbf{H}$. To see the reverse inclusion, take a point, $q \in \mathbf{H} \setminus \bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R)$. This means that q is not interior to some $\mathbf{A}_{w_\ell}(R)$ and therefore the shortest path from q to a furthest vertex in $\mathbf{P}_{w_\ell}$ uses more than R links. Hence, $\bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R) \supseteq lc(\mathbf{P}) \cap \mathbf{H}$.

When $R = 2$, $lc(\mathbf{H})$ is the area below the lowest horizontal edge of the top chain. From any point in this area it is possible to reach any other point in $\mathbf{H}$ using two links. Thus, $lc(\mathbf{H}) \supseteq lc(\mathbf{P}) \cap \mathbf{H}$ and by a similar argument as in the first part of the proof we get $\bigcap_{1 \le i \le m} \mathbf{A}_{w_i}(R) \cap lc(\mathbf{H}) = lc(\mathbf{P}) \cap \mathbf{H}$. □

4.1 Breaking Down the Problem

The previous results show that in order to find $lc(\mathbf{P}) \cap \mathbf{H}$ we need to compute the intersection $\bigcap_{w \in \mathbf{H}} \mathbf{A}_w(R)$ efficiently.

We will, in fact, solve a more general problem. We show how to compute, for an arbitrary value k, the intersection $\bigcap_{w \in \mathbf{H}} \mathbf{A}_w(k)$ in linear time (in the size of $\mathbf{P}$). This intersection is the set of points in $\mathbf{H}$, $\mathbf{H}$ being contained in $\mathbf{P}$, which can reach all the vertices of $\mathbf{P}$ with at most k links. In this section we will denote the intersection $\bigcap_{w \in \mathbf{H}} \mathbf{A}_w(k)$ as $\mathbf{R}(\mathbf{H})$. Note that the polygon $\mathbf{P}$ is arbitrary, e.g., it can be a subpolygon of the polygon for which we want to compute the link center.

To each window w in $\mathbf{H}$ we associate the value $\delta_w = k - d_w$ where $d_w = \max_{v \in \mathbf{P}_w} d(v, w)$. δ_w tells us how many free links there are in $\mathbf{H}$. So when $\delta_w = 0$, the region $\mathbf{R}(\mathbf{H})$ is at most one link away from the window w. In general, a point in $\mathbf{R}(\mathbf{H})$ is at most $\delta_w + 1$ links away from w.

If $\delta_w < 0$, then the region $\mathbf{A}_w(k)$ is empty and, hence, $\mathbf{R}(\mathbf{H})$ is also empty. On the other hand, if $\delta_w \ge 3$, then $\mathbf{A}_w(k) = \mathbf{H}$. This means that our algorithm only needs to concern itself with the windows for which $0 \le \delta_w \le 2$ since these are the only cases when the area $\mathbf{A}_w(k)$ is non-trivial.

We will divide the windows into different classes according to their value δ_w and whether they lie on east or west edges of $\mathbf{H}$. This gives six different classes (in fact seven, since the

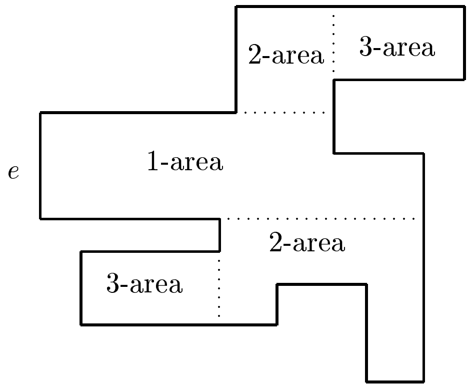

Figure 10: *The d-areas from edge e.*

base c of $\mathbf{H}$ is also considered a window. But this class has only one member, c itself). We will show how to find a constant number of regions associated to each class such that the intersection of these regions is $\mathbf{R}(\mathbf{H})$. To achieve this we need to characterize the areas $\mathbf{A}_w(k)$ properly according to the class the window w belongs to. To this end we will use the notion of the d-area.

DEFINITION 4.2 Let $\mathbf{P}$ be a simple rectilinear polygon and let e be an edge (or part of an edge) of $\mathbf{P}$. We define the *d-area* (or *visibility region*) $\mathbf{V}_{\mathbf{P}}(e, d)$ of $\mathbf{P}$ with base e recursively as

$d = 1$ the maximal histogram in $\mathbf{P}$ with e as its base,

$d > 1$ the 1-area of $\mathbf{P}$ with e as its base and the $(d-1)$-areas of the subpolygons, $\mathbf{P}_{w_1}, \ldots, \mathbf{P}_{w_m}$, cut off by the windows, $w_1, \ldots, w_m$, of $\mathbf{V}_{\mathbf{P}}(e, 1)$ with these windows as their bases. That is, $\mathbf{V}_{\mathbf{P}}(e, d) = \mathbf{V}_{\mathbf{P}}(e, 1) \cup \bigcup_{1 \le i \le m} \mathbf{V}_{\mathbf{P}_{w_i}}(w_i, d-1)$; see Figure 10.

It is easy to see that the d-area is exactly the set of points in $\mathbf{P}$ which can reach some point on e with an admissible path of length at most d.

In the following the d-area will always be interior to the maximal histogram we are looking at. This allows us to find nice properties of these areas which make them simple to work with.

We begin by giving a general lemma which characterizes the regions. Let $\mathbf{U}_w(v, k)$ be the region in $\mathbf{H}$ containing the points that can reach vertex v with at most k links. From the definition of d-area it follows that $\mathbf{U}_w(v, k)$ is equal to $\bigcup_{i=0}^{\delta_w} \mathbf{V}_{\mathbf{H}}(w(v, k-i), i+1)$.

LEMMA 4.2 *Let w be a window of $\mathbf{H}$.*

$$\mathbf{A}_w(k) = \bigcap_{v \in \mathbf{P}_w} \mathbf{U}_w(v, k).$$

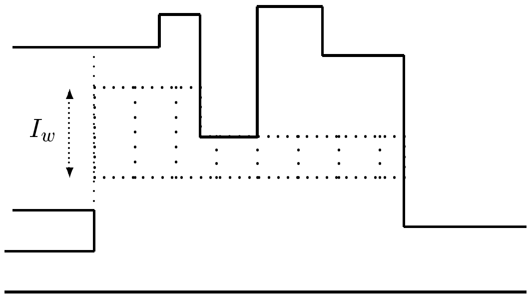

Figure 11: *The region $\mathbf{A}_w(k)$ when $\delta_w = 0$.*

PROOF: It is clear that a point in $\mathbf{U}_w(v,k)$ can reach vertex v with k links. Hence, from a point in the intersection all vertices in $\mathbf{P}_w$ can be reached with k links.

On the other hand, take a point outside the intersection, then some interval $w(v, d_w+i)$ can not be reached with $i+1$ links for any value $0 \leq i \leq \delta_w$ and hence the vertex v can not be reached with k links. $\square$

This means that we have the following equalities.

$$\mathbf{R}(\mathbf{H}) = \bigcap_{w \in \mathbf{H}} \mathbf{A}_w(k) = \bigcap_{w \in \mathbf{H}} \bigcap_{v \in \mathbf{P}_w} \mathbf{U}_w(v,k)$$

and $\mathbf{R}(\mathbf{H}) = lc(\mathbf{P}) \cap \mathbf{H}$ when $k = R$.

The rest of this section is devoted to showing that for each class of windows there is a constant number of associated regions such that the intersection of these regions equals $\mathbf{R}(\mathbf{H})$. We will look at each class separately.

4.1.1 The Windows Having One Link to $\mathbf{R}(\mathbf{H})$

We first consider the two classes when $\delta_w = 0$ and w lies on a west (east) edge. The fact that $\delta_w = 0$ implies that a point in $\mathbf{R}(\mathbf{H})$ is at most one link away from w. Let I_w be the intersection of all the intervals $w(v, d_w)$, where $v \in \mathbf{P}_w$. We get the following characterization lemma.

LEMMA 4.3 *Let w be a vertical window for which $\delta_w = 0$.*

1. $\mathbf{A}_w(k) = \emptyset$ *if* $I_w = \emptyset$,
2. $\mathbf{A}_w(k) = \mathbf{V}_{\mathbf{H}}(I_w, 1)$ *if* $I_w \neq \emptyset$.

PROOF: Follows immediately from Lemma 4.2 and the definition of visibility region (*d*-area). $\square$

See Figure 11 for an example of the area $\mathbf{A}_w(k)$ when I_w is non-empty.

The next lemma shows that in order for $\mathbf{R}(\mathbf{H})$ to be non-empty, $\mathbf{H}$ can only have one window in each of the two classes when $\delta_w = 0$.

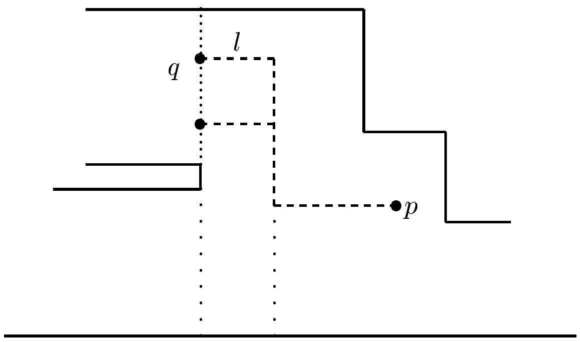

Figure 12: *p can reach any point below q using no extra link.*

Lemma 4.4 *Let w and w' be two vertical windows for which $\delta_w = \delta_{w'} = 0$ and such that both windows lie on west (east) edges of $\mathbf{H}$. The intersection of $\mathbf{A}_w(k)$ and $\mathbf{A}_{w'}(k)$ equals the empty set.*

Proof: Assume the contradiction, i.e., $\mathbf{A}_w(k) \cap \mathbf{A}_{w'}(k) \neq \emptyset$. Take a point p interior to this intersection. Then from p it is possible to reach both windows w and w' using one link. But since the windows are both on west (east) edges the same link will have to reach both windows, a contradiction. □

Next, we consider the two classes when $\delta_w = 1$. But before characterizing the areas $\mathbf{A}_w(k)$ associated to these windows we need the following lemma.

Lemma 4.5 *Let w be a vertical window and q a point on w. If a point p can reach q with an admissible path (w.r.t. w) of d links, with $d \geq 2$, any point on w below q can also be reached with an admissible path of d links from p.*

Proof: Take an admissible path $\mathcal{P}$ from p to q. The last link l of $\mathcal{P}$ is horizontal. Because of the histogram property of $\mathbf{H}$ the end points of l can be projected onto the base of $\mathbf{H}$, giving a rectangle interior to $\mathbf{H}$. One of the sides of the rectangle is the extension of the second to last link of $\mathcal{P}$ towards the base of $\mathbf{H}$. The opposite side of the rectangle contains all the points of w below q. Hence, from any point below q we can connect a link to the opposite side of the rectangle and from there continue to p using an extension of the second to last link of $\mathcal{P}$; refer to Figure 12. □

4.1.2 The Windows Having Two Links to $\mathbf{R}(\mathbf{H})$

The next lemma characterizes the regions $\mathbf{A}_w(k)$ when $\delta_w = 1$. The lemma does not give a complete characterization, but the result is sufficient for us to prove that each of the two classes can have at most two relevant windows. If there are more than two relevant windows, $\mathbf{R}(\mathbf{H})$ will be empty.

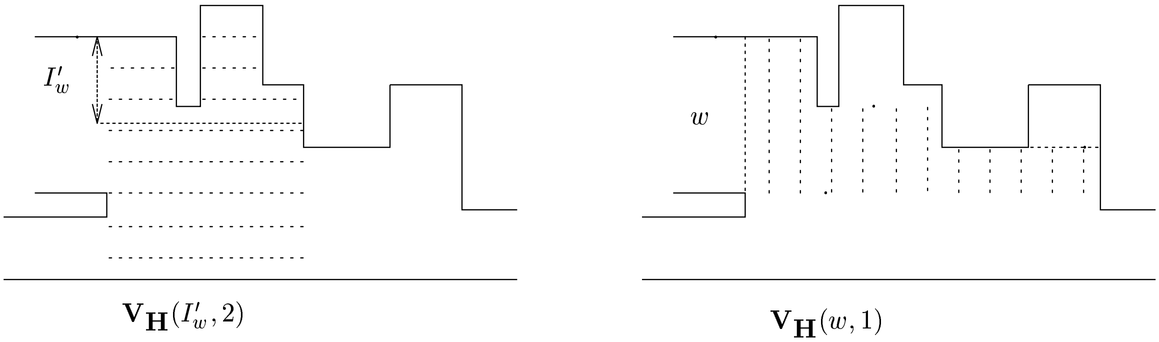

Figure 13: *Example of regions bounding $\mathbf{A}_w(k)$ when $\delta_w = 1$.*

Let I'_w be the interval which has the top end point of w as its upper end point and the topmost of the bottom end points of the intervals $w(v, d_w)$ as its lower end point.

LEMMA 4.6 *Let w be a vertical window for which $\delta_w = 1$.*

$$\mathbf{V_H}(I'_w, 2) \subseteq \mathbf{A}_w(k) \subseteq \mathbf{V_H}(I'_w, 2) \cup \mathbf{V_H}(w, 1).$$

PROOF: Take a point $p \in \mathbf{V_H}(I'_w, 2)$. Since p can reach some point on I'_w with two links (using an admissible path), it follows from Lemma 4.5 that all points below the lower end point of I'_w can also be reached with two links using admissible paths. Now, since there is at least one point from each interval $w(v, d_w)$, with $v \in \mathbf{P}_w$, in the interval $w \setminus I'_w$ (we include the bottom end point of I'_w), it is possible to reach each interval $w(v, d_w)$ from p with two links and, hence, each vertex $v \in \mathbf{P}_w$ can be reached from p with k links. Recall that $d_w = k - 1$.

To see the other inclusion, take a point $p \in \mathbf{A}_w(k) \setminus \mathbf{V_H}(I'_w, 2)$. If p can reach a point in each interval $w(v, d_w)$, where $v \in \mathbf{P}_w$, with two links then, in particular, it can reach some point on I'_w and is, therefore, in $\mathbf{V_H}(I'_w, 2)$, a contradiction. This means that there is some interval $w(v, d_w)$, where $v \in \mathbf{P}_w$, that can not be reached with two links. This, in turn, means that the path from p to v must have one link inside $\mathbf{H}$ and $d_w + 1$ links in $\mathbf{P}_w$. Hence, the interval $w(v, d_w + 1)$, where $v \in \mathbf{P}_w$, must be reached from p with one link. So, $p \in \mathbf{V_H}(w(v, d_w + 1), 1) \subseteq \mathbf{V_H}(w, 1)$, for all intervals $w(v, d_w + 1)$, where $v \in \mathbf{P}_w$. □

An example of the regions bounding $\mathbf{A}_w(k)$ when $\delta_w = 1$ is given in Figure 13.

In the next lemma we prove that some windows in the class will not further restrict the region $\mathbf{R}(\mathbf{H})$ and can therefore be disregarded.

LEMMA 4.7 *If w and w' are two vertical windows both lying on west (east) edges of $\mathbf{H}$, with $\delta_w = \delta_{w'} = 1$, such that w' lies on the boundary of $\mathbf{V_H}(I'_w, 2)$, then $\mathbf{A}_{w'}(k) \subseteq \mathbf{A}_w(k)$.*

PROOF: Assume w.l.o.g. that the windows lie on west edges.

We will prove that

$$\mathbf{A}_{w'}(k) \subseteq \mathbf{V_H}(I'_{w'}, 2) \cup \mathbf{V_H}(w', 1) \subseteq \mathbf{V_H}(w', 2) \subseteq \mathbf{V_H}(I'_w, 2) \subseteq \mathbf{A}_w(k).$$

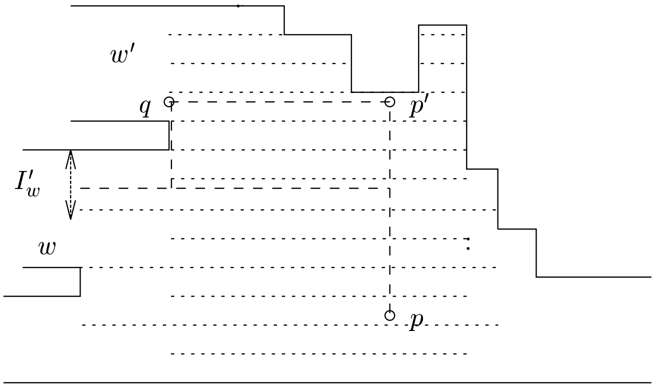

Figure 14: *p can reach both I'_w and w' with two links.*

The first inclusion follows from Lemma 4.6 and the second inclusion follows from the definition of d-area, since $I'_{w'} \subseteq w'$.

To prove the third inclusion take a point $p \in \mathbf{V_H}(w', 2)$. p can reach some point q on w' with two links (admissible w.r.t. w') and since $q \in \mathbf{V_H}(I'_w, 2)$ q can reach some point on I'_w with two links (also admissible w.r.t. w). Let p' be the turning point of the path between p and q. By the histogram property of $\mathbf{H}$ we can project points p' and q to the base giving an empty rectangle in $\mathbf{H}$. Extend the last link of the path from q to I'_w until it reaches the other side of the rectangle. From here one more link is needed to reach the point p (since the link between p and p' is vertical), giving two links in total; see Figure 14. The fourth inclusion follows again from Lemma 4.6. □

The windows which are not discarded by the previous lemma we call the *relevant* windows.

Finally, for the two classes when $\delta_w = 1$, we prove that if there are three or more relevant windows in one of the classes, then the intersection is empty.

LEMMA 4.8 *If w, w', and w'' are three vertical windows all lying on west (east) edges of $\mathbf{H}$, with $\delta_w = \delta_{w'} = \delta_{w''} = 1$, such that there is no window which lies on the boundary of the associated region of any of the other windows, then the intersection $\mathbf{A}_w(k) \cap \mathbf{A}_{w'}(k) \cap \mathbf{A}_{w''}(k)$ equals the empty set.*

PROOF: Again we assume w.l.o.g. that all windows lie on west edges.

We will prove that

$$\begin{array}{rcl} (\mathbf{V_H}(I'_w, 2) \cup \mathbf{V_H}(w, 1)) & \cap & \\ (\mathbf{V_H}(I'_{w'}, 2) \cup \mathbf{V_H}(w', 1)) & \cap & \\ (\mathbf{V_H}(I'_{w''}, 2) \cup \mathbf{V_H}(w'', 1)) & = & \emptyset, \end{array}$$

which together with Lemma 4.6 proves that the intersection of the regions associated to the windows w, w', and w'' is empty.

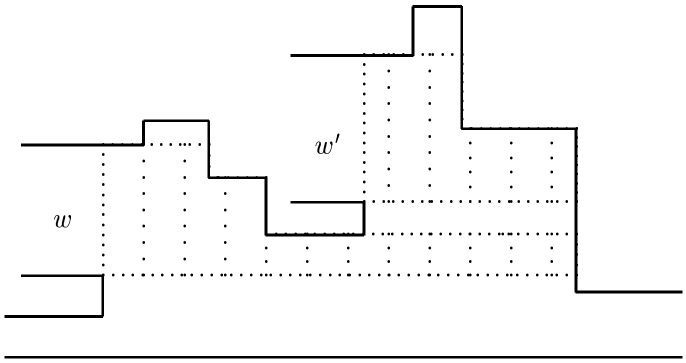

Figure 15: *The intersection of two 1-areas is empty.*

Note that if we can prove that any pair of 1-areas has an empty intersection and similarly that any pair of 2-areas has an empty intersection we are done. This follows if the above equation is rewritten in disjunctive normal form since the new formula will consist of eight terms, each being the intersection of three areas. So each term is the intersection of at least two 1-areas or two 2-areas.

Suppose that a pair of the 1-areas has a non-empty intersection. Then from a point p in this intersection we can reach the bases of the 1-areas with one link. Since both 1-areas lie on west edges, the same link from p has to reach both bases, a contradiction; see Figure 15.

We prove that a pair of 2-areas has an empty intersection by first proving that a 2-area is a histogram with a segment of the base of $\mathbf{H}$ as its base. Take a point p in the 2-area. By the histogram property of $\mathbf{H}$ we can project p onto the base. This gives an axis parallel line segment which can be extended to the top chain of $\mathbf{H}$. Any point on this segment is in the 2-area since we can follow the segment to the turning point from where p reaches a point on the corresponding window.

To prove the empty intersection of two such areas, w.l.o.g. we take the areas corresponding to w and w', assume that the window w is to the left of window w'. Project a point on w' onto the base. Since w' was not in the 2-area corresponding to window w the projected point on the base cannot be there either. But, this is the leftmost point of the 2-area of w' so the two 2-areas are disjoint; see Figure 16. □

4.1.3 The Windows Having Three Links to $\mathbf{R}(\mathbf{H})$

We look at the last two classes ($\delta_w = 2$ and w lies on a west (east) edge). We will prove that we can find the intersection of all regions having windows in these classes easily.

Again we first characterize the regions $\mathbf{A}_w(k)$ when $\delta_w = 2$, i.e., every point in $\mathbf{R}(\mathbf{H})$ is at most three links away from w. Let I''_w be the interval on w which has the top end point of w as its upper end point and the topmost of the bottom end points of the intervals $w(v, d_w + 1)$, for $v \in \mathbf{P}_w$, as its lower end point. Note the difference to the previously

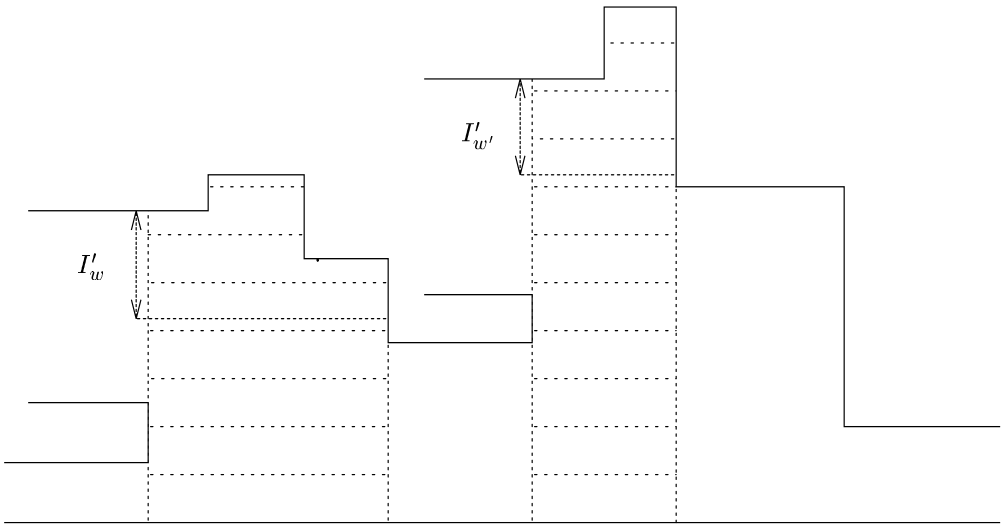

Figure 16: *The intersection of two 2-areas is empty.*

defined interval I'_w.

LEMMA 4.9 *Let w be a vertical window for which $\delta_w = 2$.*

$$\mathbf{A}_w(k) = \mathbf{V_H}(I''_w, 3).$$

PROOF: We first show that $\mathbf{A}_w(k) = \mathbf{V_H}(I'_w, 3) \cup \mathbf{V_H}(I''_w, 2)$ and then that $\mathbf{V_H}(I'_w, 3) \cup \mathbf{V_H}(I''_w, 2) = \mathbf{V_H}(I''_w, 3)$.

Take a point $p \in \mathbf{V_H}(I'_w, 3)$. Since p can reach some point on I'_w with three links (using an admissible path), it follows from Lemma 4.5 that all points below the lower end point of I'_w can also be reached with three links using admissible paths. So there is at least one point from each interval $w(v, d_w)$, on w, with $v \in \mathbf{P}_w$, which can be reached from p with three links since the interval $w \setminus I'_w$ contains at least one point from each interval $w(v, d_w)$, for every vertex $v \in \mathbf{P}_w$. Hence, all vertices in $\mathbf{P}_w$ can be reached from p with k links. Note that $d_w = k - 2$. With the same technique we can show that any point in $\mathbf{V_H}(I''_w, 2)$ is also in $\mathbf{A}_w(k)$.

To prove the equality we must show that no other points can be in $\mathbf{A}_w(k)$. Take a point $p \in \mathbf{A}_w(k) \setminus \mathbf{V_H}(I'_w, 3)$. We will show that p must be in $\mathbf{V_H}(I''_w, 2)$. The point p can not be in $\mathbf{V_H}(w, 1)$ since any path of one link from p to w can be changed to make two turns to reach a point on I'_w. Note also that the first vertical link of an admissible path from p to w of two or three links can not go upwards since then, by the histogram property of $\mathbf{H}$, I'_w is reachable from p with three links.

From the assumptions we know that there is at least one interval $w(v, d_w)$ which can not be reached from p with three links. Also, none of the intervals $w(v, d_w + 2)$ can be reached with one link and, hence, some interval $w(v, d_w + 1)$ must be reached with two

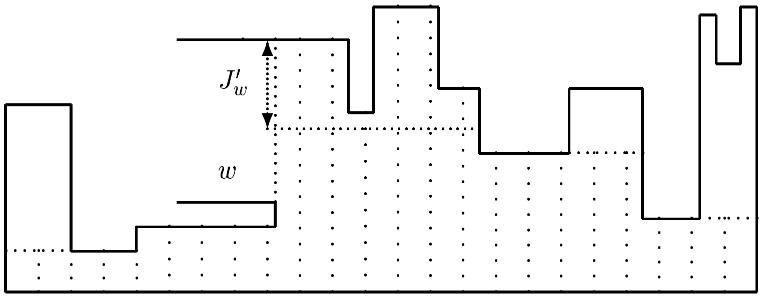

Figure 17: *The region $\mathbf{A}_w(k)$ when $\delta_w = 2$.*

links (the one for which $w(v, d_w)$ is not reachable with three links). We will show that, in fact, p can reach all the intervals $w(v, d_w + 1)$ with two links. For those intervals $w(v, d_w)$ which can be reached with two links from p the corresponding interval $w(v, d_w + 1)$ can also be reached with two links since $w(v, d_w) \subseteq w(v, d_w + 1)$. Those intervals $w(v, d_w)$ which can not be reached with two links from p can not be reached from p with three links either since the first vertical link of a path from p to $w(v, d_w)$ must go downwards and, hence, to reach $w(v, d_w)$ two new turns must be introduced in the constructed path. Hence, the corresponding intervals $w(v, d_w + 1)$ must be reachable by two links from p, and thus, all the intervals $w(v, d_w + 1)$ are reachable with two links from p, and by Lemma 4.5 the topmost of the lower end points of the intervals $w(v, d_w)$ is reachable with two links from p. This means that $p \in \mathbf{V_H}(I''_w, 2)$.

$\mathbf{V_H}(I'_w, 3) \cup \mathbf{V_H}(I''_w, 2) \subseteq \mathbf{V_H}(I''_w, 3)$ follows easily from the definition of d-area. To show the reverse inclusion take a point $p \in \mathbf{V_H}(I''_w, 3) \setminus \mathbf{V_H}(I'_w, 3)$. Note that the second to last link of a path from p to I''_w can not go from a lower point to a higher as seen from p since then, the point would be able to reach I'_w using three links. Hence, the middle link goes from a higher point to a lower. By the histogram property of $\mathbf{H}$ we can project the first link (which is horizontal) to the base of $\mathbf{H}$, giving an empty rectangle in $\mathbf{H}$. The last link connects to one side of the rectangle and can be extended to the other side. From this point, p can be reached with an additional link along the side of the rectangle. Hence, $p \in \mathbf{V_H}(I''_w, 2)$. □

See Figure 17 for an example of the region in this case.

We prove in the same manner as for the classes $\delta_w = 1$ that some windows can be disregarded.

LEMMA 4.10 *Let w and w' be two vertical windows both lying on west (east) edges of $\mathbf{H}$, with $\delta_w = \delta_{w'} = 2$, and w' lies on the boundary of $\mathbf{V_H}(I''_w, 2)$. Then, $\mathbf{A}_{w'}(k) \subseteq \mathbf{A}_w(k)$.*

PROOF: Again we assume w.l.o.g. that the windows lie on west edges.

Take a point $p \in \mathbf{A}_{w'}(k) = \mathbf{V_H}(I''_{w'}, 3)$. p can reach some point q on $I''_{w'}$ with three links and since $q \in \mathbf{V_H}(I''_w, 2)$, q can reach some point on I''_w with two links. Let p' be the last turning point of the path from p to q. By the histogram property of $\mathbf{H}$ we can project points p' and q to the base giving an empty rectangle in $\mathbf{H}$. Extend the last link of the path from q to I''_w until it reaches the other side of the rectangle. From here (at most) two more links are needed to reach the point p, giving three links in total. Hence, $p \in \mathbf{V_H}(I''_w, 3) = \mathbf{A}_w(k)$. □

Again, we call the windows that can not be discarded by the previous lemma the relevant windows.

Finally, for the two classes of windows when $\delta_w = 2$, we prove that if there is more than one relevant window, then the intersection can be found easily.

Let $w_1, \dots, w_\ell$ be ℓ vertical windows, $\ell \geq 2$, lying on west (east) edges of $\mathbf{H}$ (in the order of a scan of the boundary of $\mathbf{H}$), with $\delta_{w_1} = \cdots = \delta_{w_\ell} = 2$, such that none of the windows lie on the boundary of the corresponding 2-area of any other window.

Let e be the lowest horizontal segment on the boundary of $\mathbf{H}$ such that w_1 lies before e (in a scan along the boundary) and w_ℓ lies after e. We have the following lemma.

Lemma 4.11

$$\bigcap_{1 \leq i \leq \ell} \mathbf{A}_{w_i}(k) = \mathbf{V_H}(e, 2).$$

Proof: Again we assume w.l.o.g. that the windows lie on west edges.

Take a point $p \in \mathbf{V_H}(e, 2)$. Extend a horizontal line through p until it reaches both the left and right boundary point. If any window of the class lies below the line, it can only be w_1 since e is the lowest edge after w_1. From the other windows p can be reached by a short horizontal link and a vertical down to the horizontal line extending from p. In the same way any point on w_1 can reach p with a short horizontal link and a vertical up to the line extending from p.

To prove that no other point can be in the intersection take a point $p \in \bigcap_{1 \leq i \leq \ell} \mathbf{A}_{w_i}(k)$. Since the last link to each window is horizontal the first link from p must also be horizontal. So, any horizontal edge between the first and the last window can be reached with two links from p (specifically e). □

See Figure 18 for an example.

Finally, we look at the class consisting only of c, the base of $\mathbf{H}$. We want to show properties that enable an algorithm to compute $\mathbf{A}_c(k)$ efficiently.

4.1.4 The Region for the Base Window

We will in the following only concern ourselves with the case $\delta_c = 1$, since this is the only case when the computation gets involved. Recall that this means that $d_c = k - 1$. The other cases are treated in Lemma 4.14.

From de Berg [B89] we know that the intervals $c(v, d_c)$ and $c(v, d_c + 1)$ have one end point in common (and, of course, that $c(v, d_c + 1)$ extends at least as far as $c(v, d_c)$). So we will partition the vertices in $\mathbf{P}_c$ into two sets, $S_{\mathbf{L}}$ and $S_{\mathbf{R}}$. For any vertex $v \in S_{\mathbf{L}}$, the corresponding interval $c(v, d_c + 1)$ extends to the left of the interval $c(v, d_c)$. The set $S_{\mathbf{R}}$

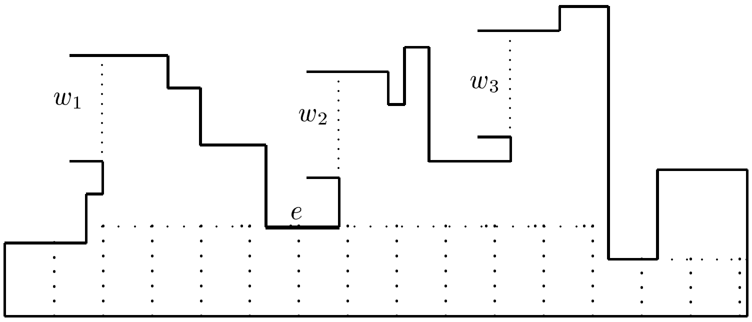

Figure 18: *The intersection when $\delta_w = 2$.*

contains the vertices for which $c(v, d_c + 1)$ extends to the right of $c(v, d_c)$. The remaining vertices (for which $c(v, d_c) = c(v, d_c + 1)$) are distributed evenly among $S_{\mathbf{L}}$ and $S_{\mathbf{R}}$.

We will establish properties of the regions corresponding to the vertices in the two sets separately. The two cases are symmetric so we will only show the properties for the set $S_{\mathbf{L}}$. Remember from the beginning of Section 4.1 that the region $\mathbf{U}_w(v, k)$ is the region associated to vertex v in $\mathbf{P}_w$ such that any point in the region can reach v with k links, i.e., $\mathbf{U}_c(v, k) = \mathbf{V_H}(c(v, k-1), 2) \cup \mathbf{V_H}(c(v, k), 1)$ in our case, when $\delta_c = 1$. We denote the area $\bigcap_{v \in S_{\mathbf{L}}} \mathbf{U}_c(v, d_c + 1)$ by $\mathbf{L}$ (again, $d_c + 1 = k$). Similarly, we define the area $\mathbf{R}$ for the vertices in $S_{\mathbf{R}}$.

An interval $c(v, d_c)$ and its corresponding interval $c(v, d_c+1)$, with $v \in S_{\mathbf{L}}$, is considered a pair or, more precisely, as one interval having a designated point in the middle (the left end point of $c(v, d_c)$). We denote the end points and the designated point as α_v, β_v, and γ_v in order from left to right, for each vertex v in $S_{\mathbf{L}}$.

Let v_l be a vertex in $S_{\mathbf{L}}$ such that there is no vertex $v \in S_{\mathbf{L}}$ with γ_v to the left of γ_{v_l}.

Lemma 4.12 *Let v and v' be two vertices, with $v, v' \in S_{\mathbf{L}}$ and such that $\alpha_{v'}$ is to the left of α_v. If $\beta_{v'}$ is to the left of β_v, then*

$$\mathbf{U}_c(v, d_c + 1) \cap \mathbf{U}_c(v_l, d_c + 1) \subseteq \mathbf{U}_c(v', d_c + 1).$$

Proof: If $\mathbf{U}_c(v, d_c+1) \cap \mathbf{U}_c(v_l, d_c+1) = \emptyset$, then the lemma is obviously true. So, assume the intersection to be non-empty. Take a point $p \in \mathbf{U}_c(v, d_c + 1) \cap \mathbf{U}_c(v_l, d_c + 1)$. We will show that from p it is possible to reach v' using at most k links.

Suppose first that p lies to the left of $\alpha_{v'}$. This means that p can not be in $\mathbf{V_H}(c(v, d_c+1), 1)$ since $\alpha_{v'}$ is to the left of α_v. Then, $p \in \mathbf{V_H}(c(v, d_c), 2)$ and by the histogram property of $\mathbf{H}$, some point to the right of $\beta_{v'}$ can be reached with two links since a point to the right of β_v can be reached with two links. Hence, $p \in \mathbf{V_H}(c(v', d_c), 2) \subseteq \mathbf{U}_c(v', d_c + 1)$.

If p lies between $\alpha_{v'}$ and $\gamma_{v'}$, then one link straight down hits the interval $c(v', d_c + 1)$ and $p \in \mathbf{V_H}(c(v', d_c + 1), 1) \subseteq \mathbf{U}_c(v', d_c + 1)$.

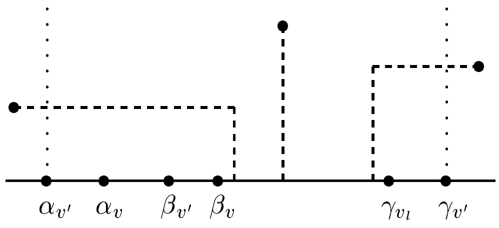

Figure 19: *The region associated to v' can be disregarded.*

Finally, if p lies to the right of $\gamma_{v'}$, then, again by the histogram property of $\mathbf{H}$, some point to the left of $\gamma_{v'}$ can be reached with two links since a point to the left of γ_{v_l} can be reached with two links, $p \in \mathbf{V_H}(c(v_l, d_c), 2)$. Remember that γ_{v_l} is the leftmost right end point of all the intervals, so all points γ_v lie to the right of γ_{v_l}. Hence, $p \in \mathbf{V_H}(c(v', d_c), 2) \subseteq \mathbf{U}_c(v', d_c + 1)$; see Figure 19. □

Let $S'_{\mathbf{L}}$ be the subset of $S_{\mathbf{L}}$ containing the vertices v for which there is no vertex v' with α_v to the left of $\alpha_{v'}$ and β_v to the left of $\beta_{v'}$. By the previous lemma it is clear that $\mathbf{L} = \bigcap_{v \in S'_{\mathbf{L}}} \mathbf{U}_c(v, d_c + 1) \cap \mathbf{U}_c(v_l, d_c + 1)$.

Next, we concentrate on the intersection $\bigcap_{v \in S'_{\mathbf{L}}} \mathbf{U}_c(v, d_c + 1)$. We will show how to construct the boundary of a region $\mathbf{L}'$ which is slightly larger than the intersection $\bigcap_{v \in S'_{\mathbf{L}}} \mathbf{U}_c(v, d_c + 1)$. This will turn out not to be a problem since when $\mathbf{L}'$ is intersected with $\mathbf{U}_c(v_l, d_c+1)$, $\mathbf{L}$ is the resulting region. We start by specifying some designated points on the top boundary chain of $\mathbf{L}'$.

Let v be a vertex in $S'_{\mathbf{L}}$ and let b_v denote the top boundary point of $\mathbf{V_H}(c(v, d_c), 2)$ above point α_v. b_v has the same y-coordinate as the lowest boundary point of the top chain of $\mathbf{H}$ between the points α_v and β_v. This follows since β_v can be reached from b_v with (at most) two links, one link to the right and one down. Conversely, no point above b_v can reach a point in $c(v, d_c)$ with two links, since the point would then either be outside $\mathbf{H}$ if b_v is on the boundary or the first link (which must be horizontal and towards the right) would cross the boundary in the interval between α_v and β_v. See Figure 20.

Now, assume that the boundary points $b_{v_1}, \ldots, b_{v_m}$, $v_j \in S'_{\mathbf{L}}$, are given (in order from left to right). $b_{v_1}, \ldots, b_{v_m}$ have increasing y-coordinate which follows from the fact that for any vertex v_j the interval $[\alpha_{v_j}, \beta_{v_j}]$ contains $[\alpha_{v_{j+1}}, \beta_{v_{j+1}}]$; see Lemma 4.12. Let $\mathcal{C}$ be a chain consisting of the following two subchains. The left subchain is the topmost monotone (w.r.t. both the x- and the y-axis) rectilinear chain which passes through all the points $b_{v_1}, \ldots, b_{v_m}$ (since the points have increasing y-coordinate). The left subchain starts at the left end point of the base of $\mathbf{H}$ and ends on the upper boundary of $\mathbf{H}$ over point β_{v_m}. The right subchain starts where the left subchain ends and follows the upper boundary of $\mathbf{H}$ to the right end point of the base.

Let $\mathbf{L}'$ be the region in $\mathbf{H}$ that lies below $\mathcal{C}$. We have the following lemma.

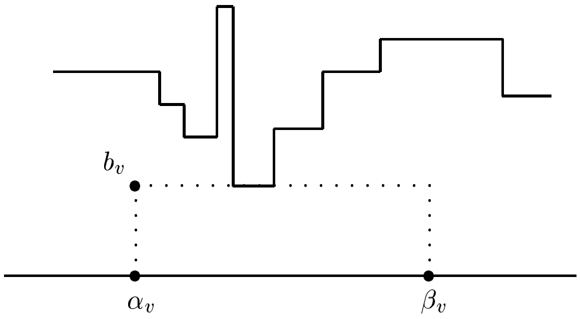

Figure 20: *The top boundary point b_v.*

LEMMA 4.13

$$\mathbf{L} = \mathbf{L}' \cap \mathbf{U}_c(v_l, d_c + 1).$$

PROOF: Take a point $p \in \mathbf{L}' \cap \mathbf{U}_c(v_l, d_c+1)$. We want to show that p is in $\mathbf{L}$. Suppose p is to the left of γ_{v_l}, to the left of β_{v_m}, and suppose also that p has the points $\alpha_{v_1}, \ldots, \alpha_{v_i}$ to the left and $\alpha_{v_{i+1}}, \ldots, \alpha_{v_m}$ to the right for some value i. Since the left subchain of $\mathcal{C}$ is monotone w.r.t. the y-axis, p is below all the points $b_{v_{i+1}}, \ldots, b_{v_m}$. Hence, p can reach all the points $\beta_{v_{i+1}}, \ldots, \beta_{v_m}$ with an admissible path of two links, i.e., the vertices $v_{i+1}, \ldots, v_m$ are reachable from p with k links. The vertices $v_1, \ldots, v_i$ are also reachable from p with k links since a link straight down from p will hit c to the left of all the points $\alpha_{v_1}, \ldots, \alpha_{v_i}$ and, hence, connect to the intervals $c(v_1, d_c+1), \ldots, c(v_i, d_c+1)$.

If p is to the right of β_{v_m} and to the left of γ_{v_l}, then a straight link down will hit the intersection of all the intervals $c(v_i, d_c)$.

If p is to the right of γ_{v_l}, then since the point γ_{v_l} is reachable by two links, any point γ_{v_i} to the left of p is reachable by two links or for the points γ_{v_i} to the right of p a straight link down will hit the corresponding intervals $c(v_i, d_c)$. Thus, we have shown that p is in $\mathbf{L}$.

To prove the reverse inclusion take a point p in $\mathbf{L}$. If p is not in $\mathbf{L}'$, then there is some interval $c(v, d_c)$ which is not reachable from p with two links and also the corresponding interval $c(v, d_c+1)$ is not reachable with one link and, hence, v is not reachable with k links from p. If p is not in $\mathbf{U}_c(v_l, d_c+1)$, then obviosly, v_l can not be reached from p with k links. This concludes the proof. □

See Figure 21 for an example of $\mathbf{L}$ and $\mathbf{L}'$.

We are now ready to characterize the region associated to the base completely. Recall that $I_c = \bigcap_{v \in \mathbf{P}_c} c(v, d_c)$, i.e., the intersection of the intervals on c reachable from the vertices in $\mathbf{P}_c$ with d_c links.

LEMMA 4.14 *Let c be the base of histogram $\mathbf{H}$. The corresponding area $\mathbf{A}_c(k)$ is*

1. $\mathbf{A}_c(k) = \emptyset$ *if $I_c = \emptyset$ and $\delta_c = 0$,*
2. $\mathbf{A}_c(k) = \mathbf{V}_{\mathbf{H}}(I_c, 1)$ *if $I_c \neq \emptyset$ and $\delta_c = 0$,*

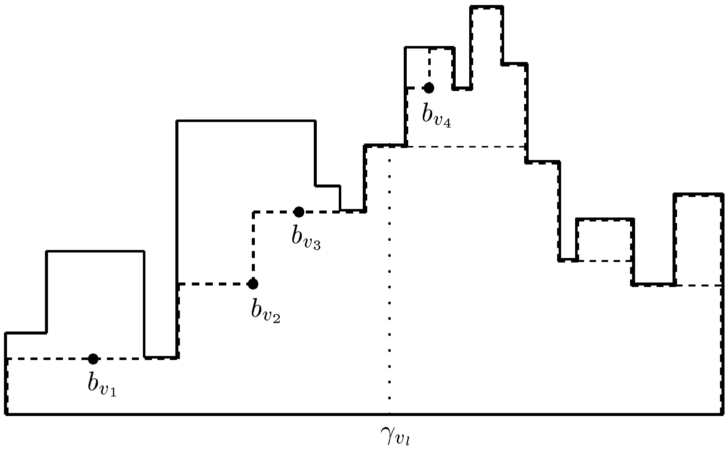

Figure 21: *The region* $\mathbf{L}$ *associated to the base when* $\delta_c = 1$.

3. $\mathbf{A}_c(k) = \mathbf{L} \cap \mathbf{R}$ *if* $\delta_c = 1$,

4. $\mathbf{A}_c(k) = \mathbf{H}$ *if* $\delta_c = 2$.

PROOF: We prove each case separately.

1. If $I_c = \emptyset$, then, for any point p on c, there exists a vertex $v \in \mathbf{P}_c$ such that $d(v,p) > k$. Hence, a path from a point in $\mathbf{H}$ passing through p will have to use more than k links to reach v, i.e., $\mathbf{A}_c(k) = \emptyset$.

2. If $I_c \neq \emptyset$, then, from any point p on I_c, every vertex in $\mathbf{P}_c$ can be reached from p with at most k links. Thus, $\mathbf{A}_c(k) \supseteq \mathbf{V_H}(I_c, 1)$. To see the reverse inclusion take a point $q \in \mathbf{H} \setminus \mathbf{V_H}(I_c, 1)$. To reach a point on I_c a path from q uses at least two links giving a total length of the path to some vertex greater than k. On the other hand, if the path does not pass through I_c then there is again a vertex $v \in \mathbf{P}_c$ that can not be reached by a path from q with less than $k+1$ links.

3. Follows from the preceding discussion.

4. Take a point $p \in \mathbf{H}$. By the histogram property p can be projected onto the base c. Thus, using two more links, any point on c can be reached from p with an admissible path and any vertex $v \in \mathbf{P}_c$ can be reached with k links.

□

4.2 Computing the Intersection $\mathbf{R}(\mathbf{H})$

In the previous section we have broken down the problem of finding the intersection $\bigcap_{w \in \mathbf{H}} \mathbf{A}_w(k) = \mathbf{R}(\mathbf{H})$ into seven different cases. We have also analyzed each of the cases in order to establish a constant number of regions that, when intersected, yield the region $\mathbf{R}(\mathbf{H})$. In this section we will concentrate on the algorithmic issues of computing $\mathbf{R}(\mathbf{H})$.

The algorithm scans the boundary of $\mathbf{H}$ and computes the corresponding value δ_w for each window; see [B89]. If there is some window w for which $\delta_w < 0$, the algorithm immediately returns the empty set since there are vertices in $\mathbf{P}_w$ that are not reachable with k links from any point in $\mathbf{H}$. If all the values are greater than or equal to 0, the windows are partitioned into seven different sets according to the classes they belong to. For each window w, the corresponding intervals I_w, I'_w, and I''_w are found as in [B89] (for the base only the interval I_c is relevant), then a constant number of regions are computed for each of the seven classes. The resulting regions are then intersected. We look at the routines which compute the regions for each class.

w is a west edge and $\delta_w = 0$

From Lemma 4.4 we know that there can be at most one window in the class. If there are two or more, the algorithm returns the empty set. If there is only one window w, the algorithm checks the interval I_w. If I_w is empty, the empty set is returned. If I_w is non-empty, the boundary of $\mathbf{H}$ is scanned in order to find the 1-area $\mathbf{V_H}(I_w, 1)$ which is returned as the resulting region.

w is an east edge and $\delta_w = 0$

This case is symmetric to the previous case.

w is a west edge and $\delta_w = 1$

From Lemma 4.7 we know that we can disregard all the windows which have a window of the class on the boundary of the associated region. The windows which can not be discarded in this way are called the relevant windows. To find the relevant windows the algorithm proceeds as follows (we may assume that the windows are given in order from left to right). Follow the top chain of $\mathbf{H}$ from left to right. Whenever a window of the class is encountered it is pushed onto a stack and whenever an east edge goes below the lower end point of the interval I'_w, where w is the top window on the stack, the window w is stored in the list containing the relevant windows and the stack is emptied. The correctness of this scheme can be easily seen.

From Lemma 4.8 we also know that in order for the intersection to be non-empty there can be at most two of the relevant windows. Hence, the algorithm will return the associated regions of at most two windows or the empty set.

One problem with this class is that a region $\mathbf{A}_w(k)$ of the class can not be specified completely in terms of 1-areas and 2-areas. It consists of two regions as follows $\mathbf{A}_w(k) = \mathbf{V_H}(I'_w, 2) \cup \mathbf{W}_w$, where $\mathbf{W}_w$ is the part of $\mathbf{A}_w(k)$ which is not in the 2-area; see Lemma 4.6. The region $\mathbf{V_H}(I'_w, 2)$ can be computed in a straightforward manner by a scan of the boundary of $\mathbf{H}$ in linear time. To compute the region $\mathbf{W}_w$ we apply a plane sweep approach

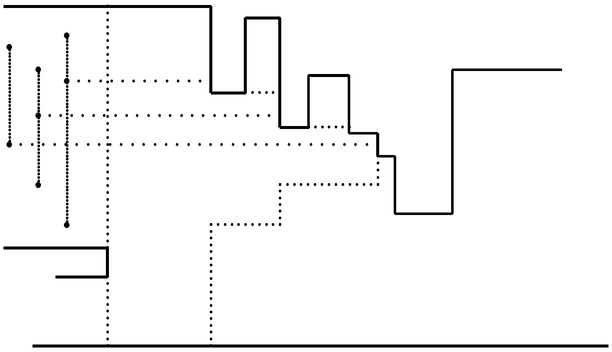

Figure 22: *The region $\mathbf{A}_w(k)$ when $\delta_w = 1$.*

from left to right. The region starts at the right vertical boundary of $\mathbf{V_H}(I'_w, 2)$ and the algorithm keeps lists of the intervals $w(v, d_w)$ and $w(v, d_w + 1)$, for all vertices $v \in \mathbf{P}_w$. The intervals $w(v, d_w)$ are sorted on the lower end point from top to bottom. The intervals $w(v, d_w+1)$ are given in the same order; see Section 3.3. During the plane sweep two points a and b on the sweep line trace the upper and lower boundary of $\mathbf{W}_w$. The upper point a has the same y-coordinate as the lowest point encountered on the top chain. The point b has the same y-coordinate as the topmost of the lower end points of the intervals $w(v, d_w + 1)$ for which the associated intervals $w(v, d_w)$ lie completely above a. see Figure 22. In the figure the horizontal dotted lines extend from the lower end points of the intervals $w(v, d_w)$. The end points that lie below are the lower end points of the intervals $w(v, d_w + 1)$. The correctness and linearity of the algorithm can be verified easily and are therefore left as exercises to the interested reader.

w is an east edge and $\delta_w = 1$

This case is symmetric to the previous case.

w is a west edge and $\delta_w = 2$

In the same way as in the case $\delta_w = 1$ some of the windows in this class can be disregarded leaving a set of relevant windows; see Lemma 4.10. The discarding is done in the same way, using a stack and moving along the upper chain of $\mathbf{H}$ from left to right.

From Lemma 4.11 we know what the intersection should be, $\mathbf{V_H}(e, 2)$. The segment e is found by a scan of the top chain of $\mathbf{H}$. It is the lowest horizontal edge having relevant windows of the class both to the left and right of it.

w is an east edge and $\delta_w = 2$

This case is symmetric to the previous case.

c is the base of H

From Lemma 4.14 we know that Cases 1 and 4 are trivial and that Case 2 can easily be computed by a scan of the top chain of $\mathbf{H}$. This leaves the hard case (number 3).

We will only show how to compute the region $\mathbf{L}$ since the region $\mathbf{R}$ can be computed using a symmetric algorithm. First, partition the vertices in $\mathbf{P}_c$ into the sets $S_\mathbf{L}$ and $S_\mathbf{R}$ and consider the intervals corresponding to the vertices of $S_\mathbf{L}$. From Lemma 4.12 we know that we can remove those vertices v' for which there is some other vertex v where $\alpha_{v'}$ lies to the left of α_v and $\beta_{v'}$ lies to the left of β_v. This is done in the following way. Keep a lists of the triples $(\alpha_v, \beta_v, \gamma_v)$, $v \in S_\mathbf{L}$. From the result in Section 3.3 we can assume that the list is sorted on the points β_v from right to left. Now, scan the list from beginning to end and look at the adjacent pairs of elements. If $\alpha_{v_{i+1}}$ is to the left of α_{v_i} then remove the element $(\alpha_{v_{i+1}}, \beta_{v_{i+1}}, \gamma_{v_{i+1}})$ from the list.

The next thing to find are the boundary points $b_{v_1}, \ldots, b_{v_m}$ of the left subchain of $\mathcal{C}$. With the list of triples we scan the boundary of $\mathbf{H}$ and scan the part between the points α_{v_m} and β_{v_m} for the lowest boundary point. The y-coordinate of the lowest boundary point together with the x-coordinate of α_{v_m} give us the point b_{v_m}. Now, assuming that we have computed b_{v_i}, then we can find the point $b_{v_{i-1}}$ as follows. Scan the part of the boundary between α_{v_i} and $\alpha_{v_{i-1}}$ as well as the part between β_{v_i} and $\beta_{v_{i-1}}$ for the lowest point. Take the minimum y-coordinate of this point and b_{v_i}, and the x-coordinate of $\alpha_{v_{i-1}}$ to get the point $b_{v_{i-1}}$. The right part of the subchain $\mathcal{C}$ is the part of the boundary of $\mathbf{H}$ to the right of b_{v_m} which can be added to the chain in a straightforward way. Thus we have computed the region $\mathbf{L}'$ in linear time. The region $\mathbf{U}_c(v_l, d_c + 1)$ is also straightforward to find. So, it remains to take the intersection of these two regions to get $\mathbf{L}$. Since they are both monotone with respect to the base of $\mathbf{H}$, this intersection can be computed in linear time. This ends the analysis for all the cases.

To see that our algorithm uses linear time in total note that, for each of the classes when $\delta_w = 0$, there can be at most one region. For the classes when $\delta_w = 1$, there can be at most two regions. When $\delta_w = 2$, we can compute the intersection in linear time for each of the two classes, and finally, for the base there are at most two regions that have to be intersected. This gives in total 10 regions that have to be intersected, but since the regions are all monotone along the base of $\mathbf{H}$ the intersection can be computed in linear time. Thus, we have proven the main result of this section.

THEOREM 2 *Let $\mathbf{P}$ be an arbitrary rectilinear polygon and $\mathbf{H}$ a maximal histogram in $\mathbf{P}$ having c, a rectilinear chord in $\mathbf{P}$, as its base. The algorithm described above computes the region inside $\mathbf{H}$ from which all other points in $\mathbf{P}$ can be reached with at most k links. The algorithm runs in linear time in the size of $\mathbf{P}$.*

In particular, we have the following corollary.

COROLLARY 3 *The region $lc(\mathbf{P}) \cap \mathbf{H}$ can be computed in linear time.*

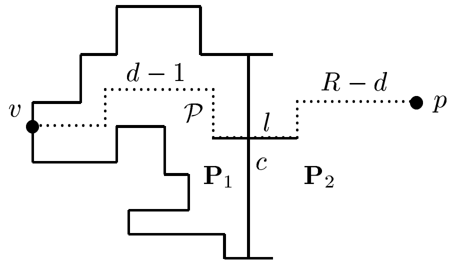

Figure 23: *v forces the part of the link center of $\mathbf{P}$ to be close to c.*

5 Computing the Link Center

We now turn to the task of computing the link center of a rectilinear polygon $\mathbf{P}$. Throughout this section we assume that we are given a chord c that splits $\mathbf{P}$ into two parts $\mathbf{P}_1$ and $\mathbf{P}_2$ which contain vertices $v_1 \in \mathbf{P}_1$ and $v_2 \in \mathbf{P}_2$ that are *both* rather "far away" from c. For our purposes far away means that the distances from v_1 and v_2 to c are at least $R-1$. Although we do not compute R explicitly, we will show in the next section how to find such a chord in linear time.

To illustrate why it is helpful to have a chord c that has vertices on both sides that are far away from it we consider the area in $\mathbf{P}_2$ that may contain the link center given a vertex v in $\mathbf{P}_1$ with distance d to c. We claim that a point p in the part of the link center in $\mathbf{P}_2$ has a distance of at most $R-d+1$ to c. To see this, consider a shortest path $\mathcal{P}$ from v to p and let l be the first link of $\mathcal{P}$ that intersects c (for illustration see Figure 23). Since p is in the link center of $\mathbf{P}$, we have $\lambda(\mathcal{P}) \leq R$. l is at least the d^{th} link of $\mathcal{P}$ as seen from v since $d(v,c) = d$. Hence, the part of $\mathcal{P}$ from c to p consists of at most $R-d$ links plus the part of l that is contained in $\mathbf{P}_2$. Therefore, we have $d(p,c) \leq R-d+1$. For further reference, we generalize the result and state it as a lemma.

LEMMA 5.1 *Let c be a chord in $\mathbf{P}$ that splits $\mathbf{P}$ into two parts $\mathbf{P}_1$ and $\mathbf{P}_2$. If p_1 is a point in $\mathbf{P}_1$ and p_2 is a point in $\mathbf{P}_2$ such that $d(p_1,c) \geq k_1$ and $d(p_1,p_2) \leq k_2$, then $d(p_2,c) \leq k_2 - k_1 + 1$.*

Hence, the larger d is, the more restricted is the area in $\mathbf{P}_2$ that may contain the link center. Since we want to split the polygon only once, we choose c in such a way that the part of the link center of $\mathbf{P}$ that is contained in $\mathbf{P}_2$ as well as the part that is contained in $\mathbf{P}_1$ is sufficiently close to c. We denote the distance from c to the furthest vertex in $\mathbf{P}_1$ by d_1 and the distance to the furthest vertex in $\mathbf{P}_2$ by d_2, i.e., $d_i = \max_{v \in \mathbf{P}_i} d(v,c)$, for $i = 1,2$. Since we require d_1 and d_2 to be both at least $R-1$, "sufficiently close" means that the points in the link center of $\mathbf{P}$ have a distance of at most $R-(R-1)+1 = 2$ links to c.

It is the aim of this section to show that once we are given c, we can compute the link center of $\mathbf{P}$ in linear time. The remaining part of this section is dedicated to proving the following theorem.

THEOREM 4 *Let c be a chord of $\mathbf{P}$ with d_1 and d_2 defined as above. If $R \leq d_i + 1$, for $i = 1, 2$, then the parts of the link center in $\mathbf{P}_1$ and $\mathbf{P}_2$ can be computed in linear time.*

5.1 Where to Look for the Link Center

So suppose we are given a chord c in $\mathbf{P}$ that meets the conditions of Theorem 4. From now on throughout this section we assume that c is vertical and that $\mathbf{P}_1$ is to the left of c and $\mathbf{P}_2$ to the right. We distinguish between two cases. (Note that $R \leq d_1 + 1$.)

(i) $R \leq d_1$,

(ii) $R = d_1 + 1$,

and the same for d_2. We will only show how to compute the part of the link center in $\mathbf{P}_2$. The part of the link center in $\mathbf{P}_1$ can, of course, be computed in the same way. As we have pointed out above, the part of the link center in $\mathbf{P}_2$ is contained in the region $\mathbf{R}$ that can be reached from c with $R - d_1 + 1$ links. If $R - d_1 + 1 \leq 1$, i.e., $R \leq d_1$, $\mathbf{R}$ is contained in the maximal histogram $\mathbf{H}_c$ of c in $\mathbf{P}_2$.

Since we know how to compute the link center of a polygon inside a histogram in linear time by Corollary 3, we can immediately solve Case (i). So consider Case (ii) and assume that $R = d_1 + 1$. From Lemma 5.1 we have that the link center is contained in the 2-area of c in $\mathbf{P}_2$ since there is a vertex v with $d(v, c) = d_1 = R - 1$ and $d(v, p) \leq R$, for all points p in the link center of $\mathbf{P}$. Unfortunately, we cannot solve the link center problem for general 2-areas. So we have to restrict the region in $\mathbf{P}_2$ that may contain the link center a bit more. But before we continue with our investigations we introduce some notation. We define $\mathcal{I}_1$ as the set of intervals $\{c(v, d_1) \mid v \in \mathbf{P}_1\}$ on c and I_1 as the intersection of all intervals in $\mathcal{I}_1$, i.e., $I_1 = \bigcap \mathcal{I}_1$. Recall that $\mathcal{I}_1$ can be computed in linear time by [B89].

First, suppose that $I_1 = \emptyset$. Let $\overline{I}_1$ be the smallest interval that intersects all the intervals in $\mathcal{I}_1$ (see Figure 24). If we consider $\overline{I}_1$ as a line segment, we can move it to the right in $\mathbf{P}_2$ until we hit a point x on the boundary of $\mathbf{P}$. Let h be the horizontal line segment from x to $\overline{I}_1$. Note that h is a chord in $\mathbf{P}_2$. Let $\mathbf{H}_a$ be the maximal histogram in $\mathbf{P}_2$ above h and $\mathbf{H}_b$ be the maximal histogram below h. We claim that if $R = d_1 + 1$ and $I_1 = \emptyset$, then the part of the link center in $\mathbf{P}_2$ is contained in $\mathbf{H}_a \cup \mathbf{H}_b \cup \mathbf{H}_c$ where $\mathbf{H}_c$ is the maximal histogram of c in $\mathbf{P}_2$ as introduced above. We state this as a lemma.

LEMMA 5.2 *If $R = d_1 + 1$ and $I_1 = \emptyset$, then $lc(\mathbf{P}) \cap \mathbf{P}_2 \subseteq \mathbf{H}_a \cup \mathbf{H}_b \cup \mathbf{H}_c$.*

PROOF: Let p be a point in the part of the link center in $\mathbf{P}_2$ that is outside $\mathbf{H}_a \cup \mathbf{H}_b$. We show that p is contained in $\mathbf{H}_c$. For illustration refer to Figure 24. h splits $\mathbf{P}_2$ into two components $\mathbf{P}_a$ and $\mathbf{P}_b$ with $\mathbf{P}_a$ being above h and $\mathbf{P}_b$ being below h. W.l.o.g. assume that p is in $\mathbf{P}_a$. Let v be a furthest neighbour of c in $\mathbf{P}_1$ with $c(v, d_1)$ below $\overline{I}_1$, i.e., below h,

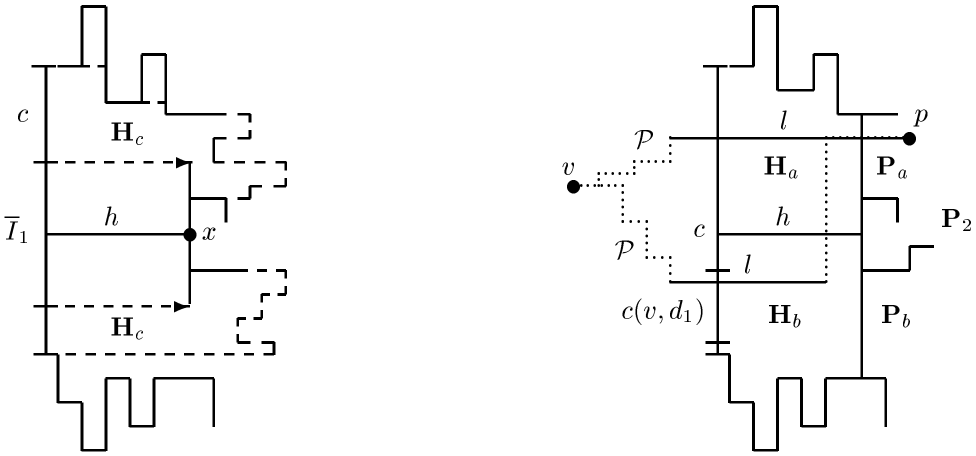

Figure 24: *$lc(\mathbf{P}) \cap \mathbf{P}_2$ is contained in $\mathbf{H}_a \cup \mathbf{H}_b \cup \mathbf{H}_c$.*

and $\mathcal{P}$ a shortest path from p to v. Let l be the first horizontal link of $\mathcal{P}$ that intersects c as seen from v. If l intersects $c(v, d_1)$, then we need at least two more links to reach p since l is below h and p is above it and outside the 1-area of h which, in particular, implies that p cannot be reached from any horizontal link below h with one link. Since l is horizontal and intersects $c(v, d_1)$, it is at least the ${d_1}^{\text{st}}$ link of $\mathcal{P}$ as seen from v (note that we have $d(v, c) = d_1$). Hence, $\mathcal{P}$ is at least $d_1 + 2$ links long which contradicts the assumption that p can reach v with a path of R links. Therefore, $\mathcal{P}$ does not intersect $c(v, d_1)$. But then l is at least the $(d_1 + 1)^{\text{st}} = R^{\text{th}}$ link of $\mathcal{P}$. By assumption, we have $\lambda(\mathcal{P}) \leq R$ and, thus, l connects to p. Since l intersects c orthogonally, p belongs to $\mathbf{H}_c$. $\square$

The above lemma implies that $lc(\mathbf{P}) \cap \mathbf{P}_2$ can be computed with three applications of the algorithm to compute the link center inside a histogram. By Corollary 3 this takes only linear time.

So we are left with the case that $R = d_1 + 1$ and I_1 is non-empty. We will show in the following that in this case the part of the link center in $\mathbf{P}_2$ is contained in the union of $\mathbf{H}_c$ and the 2-area of I_1 in $\mathbf{P}_2$ which we denote by $\mathbf{V}_{\mathbf{P}_2}(I_1, 2)$ as introduced in Section 4.

LEMMA 5.3 *If $R = d_1 + 1$ and $I_1 \neq \emptyset$, then $lc(\mathbf{P}) \cap \mathbf{P}_2 \subseteq \mathbf{V}_{\mathbf{P}_2}(I_1, 2) \cup \mathbf{H}_c$.*

PROOF: The proof is analogous to the proof of Lemma 5.2. So we again take a point p in $lc(\mathbf{P}) \cap \mathbf{P}_2$ that is not contained in $\mathbf{V}_{\mathbf{P}_2}(I_1, 2)$ and show that it belongs to $\mathbf{H}_c$. p cannot reach I_1 with 2 links; hence, there is a vertex v in $\mathbf{P}_1$ with $d(v, c) = d_1$ such that at least three links are needed to reach $c(v, d_1)$ from p. Let $\mathcal{P}$ be a shortest path from p to v. $\mathcal{P}$ cannot intersect $c(v, d_1)$ since in this case it consists of at least $d_1 - 1 + 3 > R$ links which contradicts the fact that p belongs to $lc(\mathbf{P})$. Hence, the part of $\mathcal{P}$ from v to c consists of at least $d_1 + 1 = R$ links. Therefore, the last horizontal link l of $\mathcal{P}$ that intersects c contains p and p belongs to $\mathbf{H}_c$. $\square$

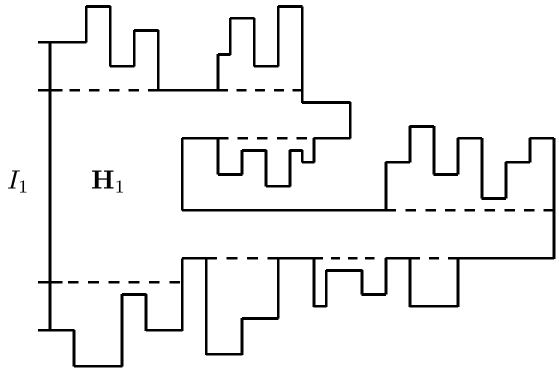

Figure 25: *The 2-area of I_1.*

We can summarize the results of this section as follows. If Case (i) holds, then we can compute the part of the link center in $\mathbf{P}_2$ in linear time making use of the previous results on how to compute the the part of the link center in a maximal histogram. In Case (ii) we can decompose the region that contains the part of the link center in $\mathbf{P}_2$ into three histograms and again use the previous results if I_1 is empty. Hence, the time spent in this case is still linear. If I_1 is non-empty, the part of the link center in $\mathbf{P}_2$ is contained in the maximal histogram $\mathbf{H}_c$ and the 2-area of I_1. Since we can compute the part of the link center in $\mathbf{H}_c$ efficiently, we will only concern ourselves with the 2-area of I_1 in the remaining part of this section. Unfortunately, it does not seem to be possible to decompose the 2-area of I_1 into a constant number of histograms so that we cannot make immediate use of Corollary 3 but, instead, we have to take a different approach.

5.2 Computing the Link Center in the 2-Area of I_1

In the remaining part of this section we show how to compute the part of the link center that is contained in the 2-area of I_1. It consists of the maximal histogram of I_1 in $\mathbf{P}_2$ which we denote in the following by $\mathbf{H}_1$ together with a number of maximal histograms that belong to the north and south windows of $\mathbf{H}_1$. An example is shown in Figure 25.

Unfortunately, our algorithm—which is presented subsequently—to compute the part of the link center in the 2-area of I_1 does not work unless we are given a vertex v in $\mathbf{P}_2$ with $d(v, I_1) \geq R$. Now, although in "most" cases such a vertex exists, it may happen that we have $d(v, I_1) < R$, for all vertices v in $\mathbf{P}_2$. In this case we split I_1 into two subintervals I_{11} and I_{12} such that $I_1 = I_{11} \cup I_{12}$ and, for each interval I_{1i}, there is a vertex v_i with $d(v_i, I_{1i}) \geq R$. We then apply our algorithm to I_{11} and I_{12} separately and, thus, obtain the part of the link center of $\mathbf{P}$ in the 2-area of I_1. The next section shows how to construct the two intervals I_{11} and I_{12}.

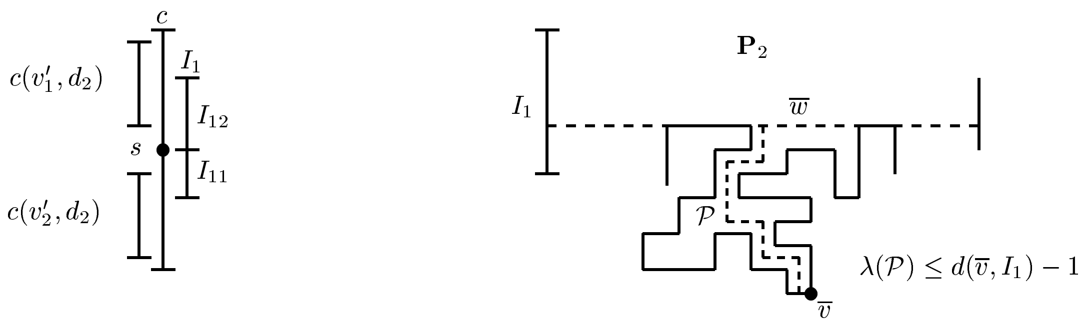

Figure 26: *The definition of* $\overline{w}$.

5.2.1 Splitting up I_1

First we investigate when there is a vertex v in $\mathbf{P}_2$ with $d(v, I_1) \geq R$. Clearly, if $d_2 \geq R$, such a vertex exists and we can apply our algorithm. So suppose that we have $R > d_2 = d_1$. We define $\mathcal{I}_2$ to be the set of intervals $\{c(v, d_2) \mid v \in \mathbf{P}_2\}$ and I_2 to be the intersection of all intervals in $\mathcal{I}_2$, i.e., $I_2 = \bigcap \mathcal{I}_2$. First assume that I_2 is non-empty. Clearly, I_1 and I_2 do not intersect since a point in the intersection can reach any other point with at most $d_1 = R - 1$ links which contradicts the fact that R is the radius. Hence, there is a vertex v in $\mathbf{P}_2$ that has distance $d_2 + 1$ to I_1.

The case that I_2 is empty is a bit more complicated. For if I_2 is empty, we cannot guarantee any more that there exists a vertex v with $d(v, I_1) \geq R$. So suppose there is no vertex in $\mathbf{P}_2$ with distance R to I_1. Since I_2 is empty, there exist two vertices v_1' and v_2' in $\mathbf{P}_2$ with $c(v_1', d_2) \cap c(v_2', d_2) = \emptyset$. Furthermore, since v_1' and v_2' can reach I_1 with d_2 links, $c(v_1', d_2)$ and $c(v_2', d_2)$ both intersect I_1.

We choose v_1' such that $c(v_1', d_2)$ has the topmost of all the lower end points of intervals in $\mathcal{I}_2$ and v_2' such that $c(v_2', d_2)$ has the bottommost of all the upper end points (see Figure 26). Clearly, v_1' and v_2' can be found in linear time. Let s be a point in I_1 between $c(v_1', d_2)$ and $c(v_2', d_2)$. We now split I_1 into one interval I_{11} below s and one interval I_{12} above s. We then have that I_{1i} does not intersect $c(v_i', d_2)$ and, hence, v_i' cannot reach I_{1i} with d_2 links, for $i = 1, 2$. In this way we obtain two intervals I_{11} and I_{12} together with two vertices v_1' and v_2' such that $I_1 = I_{11} \cup I_{12}$ and $d(v_i', I_{1i}) \geq d_2 + 1 \geq R$, for $i = 1, 2$. It is easy to see that the two intervals as well as the vertices can be found in linear time.

5.2.2 Solving the Problem for a Subinterval of I_1

The remaining problem is now the following. Given a subinterval I_1' of I_1 such that there is a vertex $\overline{v}$ in $\mathbf{P}_2$ at distance $\geq R$ to I_1' we want to find $lc(\mathbf{P}) \cap \mathbf{V}_{\mathbf{P}_2}(I_1', 2)$. Since the only property of I_1' we make use of is that all points on I_1' can reach any point in $\mathbf{P}_1$ with d_1 links, we assume for simplicity that $I_1' = I_1$. Recall that we denote the maximal histogram of I_1 in $\mathbf{P}_2$ by $\mathbf{H}_1$. Let $\overline{w}$ be the window of $\mathbf{H}_1$ that cuts off the part of $\mathbf{P}_2$ $\overline{v}$ belongs to. W.l.o.g. we assume that $\overline{w}$ is a south window. Note that $\overline{v}$ needs at least $d(\overline{v}, I_1) - 1$ links to reach $\overline{w}$ since otherwise we can reach I_1 with less than $d(\overline{v}, I_1)$ links from $\overline{v}$ (see Figure 26).

$\overline{w}$ splits $\mathbf{P}_2$ into two polygons, the polygon that is below $\overline{w}$ which we denote by $\mathbf{P}_{\overline{w}}$ and the polygon above $\overline{w}$ which we denote by $\overline{\mathbf{P}}$. Note that the only part of the 2-area of I_1 that is contained in $\mathbf{P}_{\overline{w}}$ is the maximal histogram $\mathbf{H}_{\overline{w}}$ below $\overline{w}$. Since the part of the link center in $\mathbf{H}_{\overline{w}}$ can be computed in linear time by Corollary 3, we only deal with $\overline{\mathbf{P}}$.

We first consider the case that $d(\overline{v}, I_1) > R$. We claim that the part of the link center in $\overline{\mathbf{P}}$ is then contained in the maximal histogram above $\overline{w}(\overline{v}, R)$. To see this we observe that these are the only points in $\overline{\mathbf{P}}$ that $\overline{v}$ can reach with R links. Hence, we just have to find the link center in the maximal histograms above and below $\overline{w}$. So from now on we assume that there is no point at distance $> R$ from I_1, i.e., $d(\overline{v}, I_1) = R$.

Since we cannot compute the part of the link center of $\mathbf{P}$ in the 2-area of I_1 directly, we have to find out which parts do not have to be considered. In the following we show that there are a number of histograms in the 2-area of I_1 that cannot intersect the link center of $\mathbf{P}$. So what are the parts of the 2-area of I_1 in $\overline{\mathbf{P}}$ that we do not have to take into account? The following lemma gives a first answer.

LEMMA 5.4 *The part of the link center in* $\overline{\mathbf{P}}$ *is contained in the 2-area of* $\overline{w}$.

PROOF: Let p be a point in the part of the link center of $\mathbf{P}$ in $\overline{\mathbf{P}}$. Since as we noted above $d(\overline{v}, \overline{w}) \geq R - 1$ and $d(\overline{v}, p) \leq R$, Lemma 5.1 yields that $d(p, \overline{w}) \leq 2$. □

An immediate consequence of Lemma 5.4 is that we can disregard all the maximal histograms that belong to a south window of $\mathbf{H}_1$ and also those that belong to a north window of $\mathbf{H}_1$ which cannot be reached with one link from $\overline{w}$. If we denote the maximal histogram of window w by $\mathbf{H}_w$, this can be stated as follows.

COROLLARY 5.5 *If* $\mathbf{H}_w$ *is the maximal histogram of a south or north window* w *that cannot be reached from* $\overline{w}$ *with one link, then no point in* $\mathbf{H}_w \setminus w$ *belongs to the link center of* $\mathbf{P}$.

So we only have to consider the north windows that intersect the maximal histogram $\overline{\mathbf{H}}$ above $\overline{w}$. Since we can compute the part of the link center in the maximal histogram $\overline{\mathbf{H}}$ above $\overline{w}$ in linear time, we only concern ourselves with windows w that are not completely contained in $\overline{\mathbf{H}}$. We say that window w of the maximal histogram $\mathbf{H}_1$ of I_1 is *partially visible* from $\overline{w}$ in this case. Note that there can be only one partially visible north window $\hat{w}$ whose left end point extends to the left of $\overline{w}$. Since we can compute the part of the link center in $\hat{w}$ in linear time, we assume from now on that all partially visible north windows do not extend to the left of $\overline{w}$. We denote the family of these windows by $\mathcal{W}$.

So let w be a window in $\mathcal{W}$ whose left end point can be reached with one link from $\overline{w}$. In the following we investigate which vertices of $\mathbf{P}_2$ can reach $\mathbf{H}_w$ with R links. The idea is to divide the vertices of $\mathbf{P}_2$ into five classes $\mathcal{C}_1^w, \ldots, \mathcal{C}_5^w$ which depend on w. In order

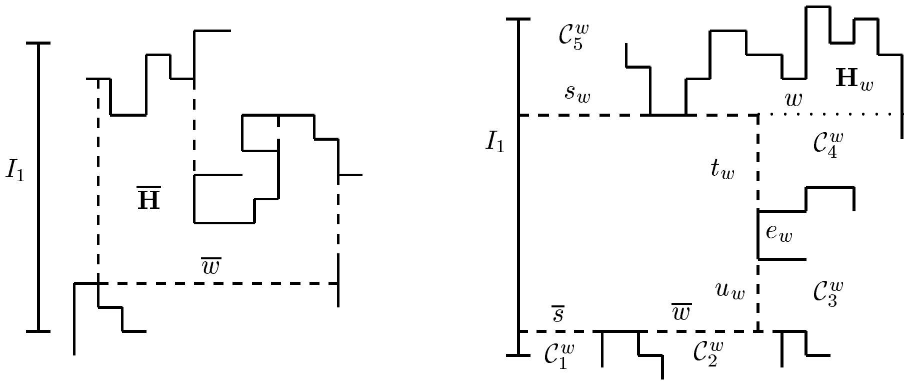

Figure 27: *The definition of* $\overline{\mathbf{H}}$, e_w, *and* $\mathcal{C}_1^w, \ldots, \mathcal{C}_5^w$.

to define these classes we denote the leftmost vertical edge of $\mathbf{H}_1$ that is below w by e_w (for illustration refer to Figure 27). Since w is partially visible from $\overline{w}$, e_w has two reflex vertices and can be reached from w with one link. Let t_w be the chord in $\mathbf{H}_1$ from the upper end point of e_w to w and u_w be the chord from the lower end point of e_w to $\overline{w}$. Furthermore, let s_w be the horizontal line segment from the left end point of w to I_1 and $\overline{s}$ be the horizontal line segment from the left end point of $\overline{w}$ to I_1. The classes of vertices are now defined as follows. $\mathcal{C}_1^w$ is defined as the class of all vertices that are below $\overline{s}$. $\mathcal{C}_2^w$ is defined as the class of all vertices below $\overline{w}$. $\mathcal{C}_3^w$ is defined as the class of vertices that are to the right of u_w and above $\overline{w}$. $\mathcal{C}_4^w$ is the class of vertices that are either to the right of t_w or above w. Finally, $\mathcal{C}_5^w$ is the class of points above s_w.

The first step in computing the part of the link center in $\mathbf{H}_w$ is to show that either *all* the points in $\mathbf{H}_w$ can reach the vertices in $\mathcal{C}_1^w$ with R links or *none*. A similar statement is true for the classes $\mathcal{C}_3^w$ and $\mathcal{C}_5^w$. We will use this fact to compute a set of windows $w_1, \ldots, w_k$ such that all the points in $\mathbf{H}_{w_i}$ can reach the vertices in $\mathcal{C}_1^{w_i}$, $\mathcal{C}_3^{w_i}$ and $\mathcal{C}_5^{w_i}$ with R links, $1 \leq i \leq k$.

In a second step we then compute the region inside each $\mathbf{H}_{w_i}$ that can be reached from the vertices in $\mathcal{C}_2^{w_i}$ and $\mathcal{C}_4^{w_i}$ with R links. This then yields the link center.

In the following let w be a window in $\mathcal{W}$. We first consider the vertices in $\mathcal{C}_1^w$. If there is a vertex v in $\mathcal{C}_1^w$ such that $d(v, \overline{s}) \geq R-1$, then the maximal histogram $\mathbf{H}_{\overline{s}}$ above $\overline{s}$ does not intersect w since we assume that the left end point of w is to the right of the left end point of $\overline{w}$. If we apply Lemma 5.4 to $\mathbf{H}_{\overline{s}}$, we obtain that $\mathbf{H}_w$ does not contain a point of the link center of $\mathbf{P}$. Since w is an arbitrary partially visible north window, we can disregard all windows in $\mathcal{W}$ and we only have to consider $\hat{w}$ which can be done in linear time by Theorem 3.

On the other hand, suppose that we have $d(v, \overline{s}) \leq R-2$, for all vertices v in $\mathcal{C}_1^w$. We consider a point p in $\mathbf{H}_w$ and a vertex v in $\mathcal{C}_1^w$. Clearly, the horizontal line segment s_w from the left end point of w to I_1 intersects the maximal histogram $\mathbf{H}$ above $\overline{s}(v, R-2)$. Note that v can reach all points in $\mathbf{H}$ with at most $R-2$ links. Since p can reach any point on

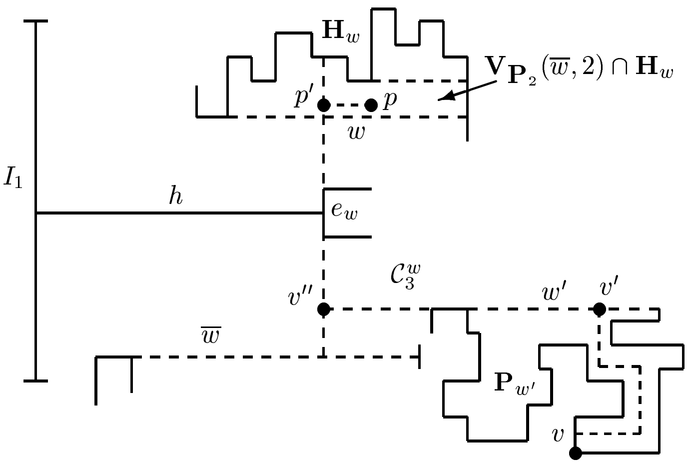

Figure 28: *If p is in the link center of $\mathbf{P}$, then v can reach w' with $R-3$ links.*

s_w with two links, we have $d(v,p) \leq R-2+2 = R$. This implies that p can reach any vertex in $\mathcal{C}_1^w$ with R links. Since w is an arbitrary partially visible north window, this is true for all windows in $\mathcal{W}$. Hence, we can disregard the vertices in $\mathcal{C}_1^w$ in this case.

Next we turn to the vertices in $\mathcal{C}_3^w$. So let v be a vertex in $\mathcal{C}_3^w$ and let w' be the window of $\mathbf{H}_1$ that cuts off the subpolygon $\mathbf{P}_{w'}$ of $\mathbf{P}_2$ which contains v. The next lemma shows that $\mathbf{H}_w \setminus \overline{\mathbf{H}}$ contains a part of the link center only if v is close enough to w'. Recall that $\overline{\mathbf{H}}$ is the maximal histogram above $\overline{w}$ and that the link center in $\mathbf{P}_2$ is contained in the 2-area of $\overline{w}$.

LEMMA 5.6 *A point p in the intersection of $\mathbf{H}_w \setminus \overline{\mathbf{H}}$ and the 2-area of $\overline{w}$ reaches v with R links if and only if $d(v,w') \leq R-3$.*

PROOF: We first show that if $p \in \mathbf{H}_w \setminus \overline{\mathbf{H}}$ is in the link center of $\mathbf{P}$, then v can reach w' with $R-3$ links. Since all the points in $\mathbf{H}_w$ that are to the left of t_w can be reached with one link from $\overline{w}$, p is to the right of t_w. Furthermore, w' is to the right of u_w since we chose w' to be in $\mathcal{C}_3^w$ (see Figure 28). Let h be the horizontal line segment from some point on e_w to c. Clearly, w' is below h. Therefore, a shortest path from p to w' has at least two horizontal links, one that intersects t_w and one that intersects u_w; thus, an admissible path from p to w' consists of at least four links. Now let v be a vertex in $\mathbf{P}_{w'}$. Since $d(p,w') \geq 4$ and $d(p,v) \leq R$, Lemma 5.1 implies that $d(v,w') \leq R-3$.

On the other hand, suppose v can reach w' with $R-3$ links. We show that $d(p,v) \leq R$ in this case. p belongs to the 2-area of $\overline{w}$; hence, there is path $\mathcal{P}$ of length 2 from a point on $\overline{w}$ to p. Since p is to the right of t_w and cannot reach $\overline{w}$ with one link, the last link of $\mathcal{P}$ is horizontal and intersects t_w in a point p'. Now let v' be some point on w' that v can reach with $R-3$ links. We project v' orthogonally onto u_w (which is always possible since we are in a histogram) and call the projection v''. The path from v' to v'' to p' to p consists of three links; hence, p can reach v with R links. □

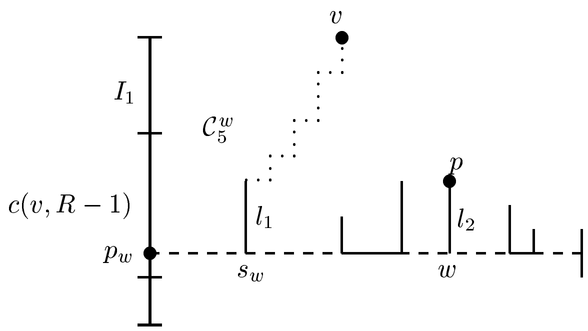

Figure 29: *v is above p.*

Let w' be the lowest window of $\mathbf{H}_1$ with $d(v, w') > R - 3$, for some vertex $v \in \mathbf{P}_{w'}$. By the above lemma we can disregard the points that are in $\mathbf{H}_w \setminus \overline{\mathbf{H}}$ of windows w in $\mathcal{W}$ where e_w is above w'. On the other hand, consider a window w in $\mathcal{W}$ such that e_w is below w' and let p be a point in the maximal histogram $\mathbf{H}_w$ of w. w' is the lowest window that cuts off a vertex with distance greater than $R - 3$ and, hence, the above lemma implies that p can reach all the vertices in $\mathcal{C}_3^w$ with R links. Since we can find all the edges e_w, where w is a window of $\mathbf{H}_1$, in time $O(|\mathbf{H}_1|)$, the windows w in $\mathcal{W}$ with e_w above w' can be computed in linear time. These will not be considered further.

In our next step we show that not only the vertices in $\mathcal{C}_3^w$ may lead to the removal of $\mathbf{H}_w$ from the possible locations of points of the link center of $\mathbf{P}$ but also that the vertices in $\mathcal{C}_5^w$ may yield that no point in $\mathbf{H}_w$ can belong to $lc(\mathbf{P})$.

Lemma 5.7 *A point p in the maximal histogram $\mathbf{H}_w$ of north window w can reach all the vertices in $\mathcal{C}_5^w$ with R links if and only if all the intervals $c(v, R-1)$, with $v \in \mathcal{C}_5^w$, contain the left end point p_w of s_w (on c).*

Proof: Let v be some vertex in $\mathcal{C}_5^w$ (see Figure 29). First assume that $c(v, R-1)$ contains p_w. Hence, there is an admissible path $\mathcal{P}$ of length $R - 1$ from v to p_w. Let l_1 be the first vertical link that intersects s_w as seen from v. Since the last link of $\mathcal{P}$ is horizontal, l_1 is at most the $(R-2)^{\text{nd}}$ link of $\mathcal{P}$. Now, p can reach any point on s_w with two links. Therefore, we can connect p to $\mathcal{P}$ yielding a path of length $R - 2 + 2 = R$.

Now suppose that there is a path $\mathcal{P}$ of length R from p to v. At least two vertical links l_1 and l_2 of $\mathcal{P}$ intersect s_w. Hence, the first vertical link l_1 that intersects s_w (as seen from v) is again at most the $(R-2)^{\text{nd}}$ link of $\mathcal{P}$. Since the intersection point of l_1 and s_w can be connected to p_w with one more horizontal link, we have $d(v, p_w) \leq R - 1$ and $p_w \in c(v, R-1)$. □

It is not too hard to see how to find the partially visible north windows of $\mathbf{H}_1$ that satisfy the conditions of Lemma 5.7 in linear time. But for completeness we give a concise description of the algorithm which works as follows. We compute a list $\mathcal{L}$ of the intervals

$c(v, R-1)$, for all vertices v in $\mathbf{P}_2$. By Section 3.3 we can assume that the list is sorted according to descending lower end points. We now process the windows in $\mathcal{W}$ from bottom to top.

Let w be the currently considered window. First we remove all the intervals from $\mathcal{L}$ that belong to vertices that are not contained in $\mathcal{C}_5^w$. Since have that $\mathcal{C}_5^{w'}$ is contained in $\mathcal{C}_5^w$ if w' is above w and we process the windows from bottom to top, we only have to remove each interval once. We then check whether the first interval in $\mathcal{L}$ contains the left end point p_w of s_w. If so, all the other intervals also contain p_w since they belong to vertices *above* s_w, i.e., above p_w, and the first element of $\mathcal{L}$ is the interval with the highest lower end point; hence, if its end point is below p_w, so are all the others. Therefore, all the vertices above s_w can reach p_w with $R-1$ links in this case. The above lemma then ensures that all vertices in $\mathcal{C}_5^w$ can reach every point in $\mathbf{H}_w$ with R links. On the other hand, if p_w is not contained in the first interval of $\mathcal{L}$, then we can remove $\mathbf{H}_w$ from the list of possible locations of the link center. If we implement $\mathcal{L}$ as a doubly-linked list and keep a reference at each vertex to its associated interval, the above procedure can be carried out in a number of steps that is proportional to the number of vertices and windows we consider.

Thus, we obtain a set of north windows $w_1, \ldots, w_k$ with the following properties. The north windows $w_1, \ldots, w_k$ are partially visible to $\overline{w}$ and a point in the maximal histogram $\mathbf{H}_{w_i}$ above w_i can reach all the vertices that belong to $\mathcal{C}_1^w$, $\mathcal{C}_3^w$, and $\mathcal{C}_5^w$ with R links.

The link center can be viewed as the region that all vertices can reach with R links. Above we have shown that the vertices of three classes can reach the points in $\mathbf{H}_{w_i}$ with R links, $1 \leq i \leq k$. So in the remaining part of this section we have to show how to compute the regions of points in $\mathbf{H}_{w_i}$ that can be reached by the vertices in $\mathcal{C}_2^w$ and $\mathcal{C}_4^w$ with R links. Taking the intersection of these two regions then yields the part of the link center in $\mathbf{H}_{w_i}$.

We first consider the vertices $\mathcal{C}_2^w$. To this end let v_1 be a vertex in the part $\mathbf{P}_{\overline{w}}$ of $\mathbf{P}_2$ below $\overline{w}$ whose interval $\overline{w}(v_1, R-1)$ has the leftmost right end point and v_2 a vertex in $\mathbf{P}_{\overline{w}}$ whose interval $\overline{w}(v_2, R-1)$ has the rightmost left end point of all intervals $\overline{w}(v, R-1)$ with v in $\mathbf{P}_{\overline{w}}$. We have the following result.

LEMMA 5.8 *A point p in $\mathbf{H}_{w_i}$ reaches all the vertices below $\overline{w}$ with R links if and only if it reaches $\overline{w}(v_1, R-1)$ and $\overline{w}(v_2, R-1)$ with two links.*

PROOF: Let p be a point in $\mathbf{H}_{w_i}$. Let I be the interval on $\overline{w}$ p reaches with an admissible path of two links with left end point q_1 and right end point q_2 (for illustration see Figure 30). To show the "if" direction we assume that $\overline{w}(v_1, R-1)$ and $\overline{w}(v_2, R-1)$ intersect I. Then, all the right end points of the $(R-1)$-intervals of vertices below $\overline{w}$ are to the right of q_1 and the left end points are to the left of q_2. This implies that I intersects all the $(R-1)$-intervals of vertices below $\overline{w}$. Hence, p can reach all vertices below $\overline{w}$ with a path of length R. If p cannot reach one of $\overline{w}(v_1, R-1)$ or $\overline{w}(v_2, R-1)$ with two links, then the shortest path from p to either v_1 or v_2 is longer than R. □

We now describe how to compute the 2-area of $\overline{w}(v_1, R-1)$ in $\mathbf{H}_{w_i}$. Clearly, the part w_i^1 of w_i that can be reached from $\overline{w}(v_1, R-1)$ with one link can be computed in linear time using the information provided by the edges e_{w_i} (see Figure 31). All the points in $\mathbf{H}_{w_i}$

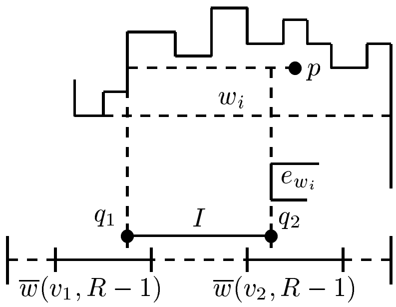

Figure 30: *I intersects $\overline{w}(v_1, R-1)$ and $\overline{w}(v_2, R-1)$.*

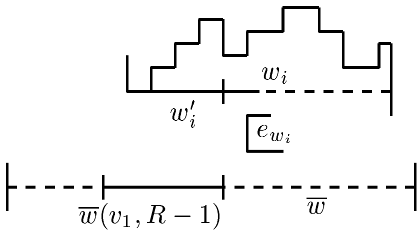

Figure 31: *The 2-area of w_i^1 can be used to compute the 2-area of $\overline{w}(v_1, R-1)$ in $\mathbf{H}_w$.*

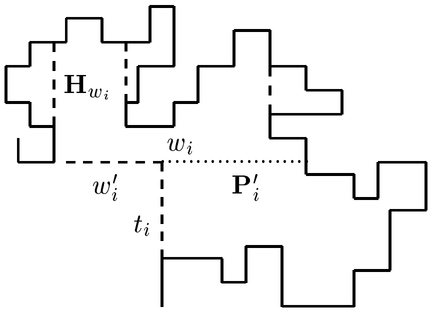

Figure 32: $\mathcal{C}_4^{w_i}$ *contains the remaining vertices.*

that can reach $\overline{w}(v_1, R-1)$ with two links belong to the 2-area of w_i^1 in $\mathbf{H}_{w_i}$ which can be computed in time proportional to the size of $\mathbf{H}_{w_i}$. Since the 2-area of a subedge of the base in a histogram is monotone, we can compute the intersection of the 2-areas of $\overline{w}(v_1, R-1)$ and $\overline{w}(v_2, R-1)$ in $\mathbf{H}_{w_i}$ also in time proportional to the number of vertices in $\mathbf{H}_{w_i}$. Hence, the total amount of time spent for computing the regions $\mathbf{R}_i$ in all $\mathbf{H}_{w_i}$, $i = 1, \ldots, k$, that can be reached by all vertices below $\overline{w}$ with R links is $O(\sum_{i=1}^{k} |\mathbf{H}_{w_i}|) = O(n)$.

Finally, we have to consider the vertices in $\mathcal{C}_4^w$, for each w_i, $i = 1, \ldots, k$. Let t_i be the chord from the upper end point of e_{w_i} to w_i and w_i' the part of w_i from t_i to the left. The part $\mathbf{P}_i'$ of $\mathbf{P}_2$ that is cut off by $t_i \cup w_i'$ just contains the vertices of $\mathcal{C}_4^{w_i}$ (see Figure 32). Hence, if we view $\mathbf{P}_i'$ as a simple polygon, we can apply Theorem 2 and compute the part $\mathbf{R}_i'$ of $\mathbf{H}_{w_i}$ that can be reached by all vertices in $\mathcal{C}_4^{w_i}$ with R links in time proportional to $|\mathcal{C}_4^{w_i}|$. Again the resulting region $\mathbf{R}_i'$ is monotone and of size $O(|\mathcal{C}_4^{w_i}|)$. In order to estimate the time needed to compute all the regions $\mathbf{R}_i'$, $i = 1, \ldots, k$ we need the following lemma.

LEMMA 5.9 *If w and w' are two windows in $\mathcal{W}$, then $\mathcal{C}_4^w$ and $\mathcal{C}_4^{w'}$ are disjoint.*

PROOF: W.l.o.g. suppose that w is above w'. This immediately implies that e_w is to the left of $e_{w'}$. Since there is no partially visible window in the part of $\mathbf{H}_1$ to the right of the chord t_w above e_w, the chord $t_{w'}$ above $e_{w'}$ is completely contained in the part of $\mathbf{P}_2$ that is to the left of t_w. Hence, $\mathcal{C}_4^w$ and $\mathcal{C}_4^{w'}$ are disjoint. □

Therefore, the amount of time spent for this step is

$$O(\sum_{i=1}^{k} |\mathcal{C}_4^{w_i}|) = O(|\mathbf{P}_2|) = O(n).$$

So we are left with the task of intersecting the regions $\mathbf{R}_i$ and $\mathbf{R}_i'$ both of which are monotone w.r.t. to x-axis and of size $O(|\mathcal{C}_4^{w_i}|)$, for $i = 1, \ldots, k$. This yields the part of the link center of $\mathbf{P}$ that is contained $\mathbf{H}_{w_i}$. Since intersecting two monotone polygons requires only time linear in the size of the polygons, this lasts step again takes time $O(\sum_{i=1}^{k} |\mathcal{C}_4^{w_i}|) = O(n)$ which completes the algorithm and shows that the part of the link center in the 2-area of I_1 can be computed in linear time.

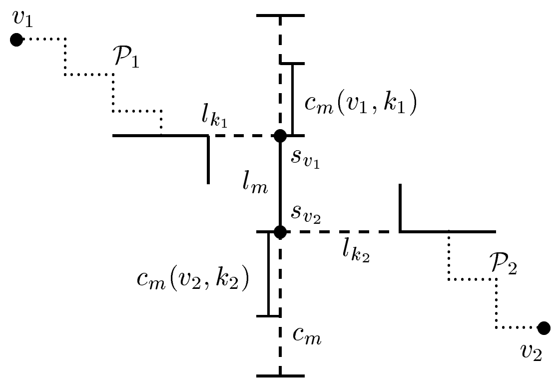

Figure 33: *l_m is the middle link of $\mathcal{P}$.*

6 Finding a Splitting Chord

In this section we show how to find a chord c in $\mathbf{P}$ that splits $\mathbf{P}$ into two parts $\mathbf{P}_1$ and $\mathbf{P}_2$ such that $\max_{v \in \mathbf{P}_1} d(v, c)$ and $\max_{v \in \mathbf{P}_2} d(v, c)$ differ by at most one from the radius R of $\mathbf{P}$. More precisely, we show the following theorem.

THEOREM 5 *Let $\mathbf{P}$ be a simple rectilinear polygon. We can find a chord c in $\mathbf{P}$ in time linear in the number of vertices of $\mathbf{P}$ that splits $\mathbf{P}$ into two parts $\mathbf{P}_1$ and $\mathbf{P}_2$ such that the radius of $\mathbf{P}$ is at most one bigger than each of $\max_{v \in \mathbf{P}_1} d(v, c)$ and $\max_{v \in \mathbf{P}_2} d(v, c)$.*

Given c, Theorem 4 of the previous section enables us to find the part of the link center in $\mathbf{P}_1$ and $\mathbf{P}_2$ in linear time. A good candidate for c is the chord c_m through the middle link l_m of a path between two vertices that span the diameter D of $\mathbf{P}$. So let v_1 and v_2 be two vertices with $d(v_1, v_2) = D$ and let $\mathcal{P}$ be a shortest path between them. We set $k_1 = \lfloor D/2 \rfloor$ and $k_2 = D - (k_1 + 1)$. This enables us to treat the cases that D is even and that D is odd in a uniform way.

Our first aim is to show that we can find a chord c with $d(v_1, c) = k_1$ and $d(v_2, c) = k_2 + 1$. To this end let l_m be the $(k_1 + 1)^{\text{st}}$ link of $\mathcal{P}$ as seen from v_1 and c_m the maximal chord through l_m. Note that l_m is also the $(k_2 + 1)^{\text{st}}$ link of $\mathcal{P}$ as seen from v_2. We have $d(v_1, c_m) = k_1$ and $d(v_2, c_m) = k_2$. As usual we assume that c_m is vertical and, further, that $c_m(v_1, k_1)$ is above $c_m(v_2, k_2)$ (see Figure 33). $c_m(v_1, k_1)$ and $c_m(v_2, k_2)$ do not intersect since otherwise we have that $d(v_1, v_2) = k_1 + k_2 - 1 < D$ by Lemma 2.3. Let s_{v_1} be the bottommost point in $c_m(v_1, k_1)$ and s_{v_2} the topmost point in $c_m(v_2, k_2)$. Furthermore, let $\mathcal{P}_1$ be a shortest path from v_1 to s_{v_1} and $\mathcal{P}_2$ a shortest path from v_2 to s_{v_2}, i.e., we have $\lambda(\mathcal{P}_1) = k_1$ and $\lambda(\mathcal{P}_2) = k_2$. W.l.o.g. suppose that l_m is the line segment from s_{v_1} to s_{v_2} and that v_1 is to the left of c_m.

In order to find c we move l_m to the left until we hit a point on the boundary of $\mathbf{P}$. We denote the new line segment by l'_m. Note that l'_m still connects to the last link l_{k_1} of $\mathcal{P}_1$ and

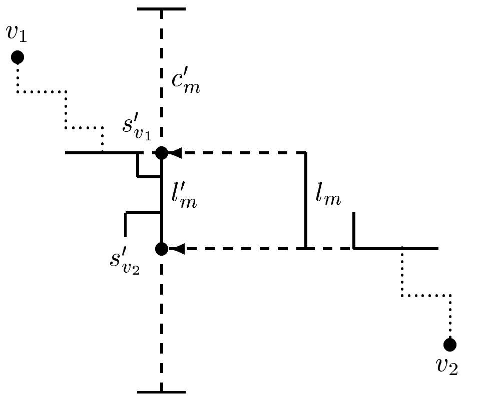

Figure 34: *The definition of l'_m and c'_m.*

that we can prolong the last link l_{k_2} of $\mathcal{P}_2$ so that l'_m also connects to $\mathcal{P}_2$ (see Figure 34). Let c'_m be the maximal line segment through l'_m and s'_{v_1} the upper end point of l'_m and s'_{v_2} the lower.

Since we allow more than two vertices of $\mathbf{P}$ to be collinear, c'_m may consist of a number of windows $w_1, \ldots, w_j$ which cut off the polygons $\mathbf{P}_{w_1}, \ldots, \mathbf{P}_{w_j}$ to the left of c'_m (for illustration refer to Figure 35). v_1 belongs to one of these polygons, say to $\mathbf{P}_{w_i}$. We now move the window w_i some distance to the left such that w_i does not intersect any vertex and is still to the right of the left end point of l_{k_1}; we denote the chord we obtain by c. Let $\mathbf{P}_1$ be the part of $\mathbf{P}$ to the left of c and $\mathbf{P}_2$ the part to the right of c. Note that w_i is above s'_{v_2} since there is at least one boundary point on l'_m.

In the following we will show that $d(v_1, c) = k_1$ and that $d(v_2, c) \geq k_2 + 1$. This is stated in the following two lemmas.

LEMMA 6.1 *If c is defined as above, then we have that $d(v_1, c) = k_1$ and that v_1 is to the left of c.*

PROOF: Our first claim is that $d(v_1, c) = k_1$. To see this assume the contrary. Then, there is a path from v_1 to c that consists of $k_1 - 1$ or less links. Hence, v_1 can reach s'_{v_2} with k_1 or less links which results in a path of length $k_1 + k_2 < D$ from v_1 to v_2. We also claim that v_1 is to the left of c. Note that the left end point of l_{k_1} is to the left of c. So if v_1 is to the right of c, then a shortest path $\mathcal{P}$ from v_1 to the left end point of l_{k_1} crosses c. The link of $\mathcal{P}$ that crosses c is at most the $(k_1 - 1)^{\text{st}}$ link as seen from v_1 which contradicts $d(v_1, c) = k_1$. $\square$

LEMMA 6.2 *With the definitions as above we have that $d(v_2, c) \geq k_2 + 1$ and that v_2 is to the right of c.*

PROOF: First we show that v_2 is to the right of c. In order to do so we first observe that v_2 cannot reach any point on c'_m that is above s'_{v_2} with k_2 or less links. For, assume

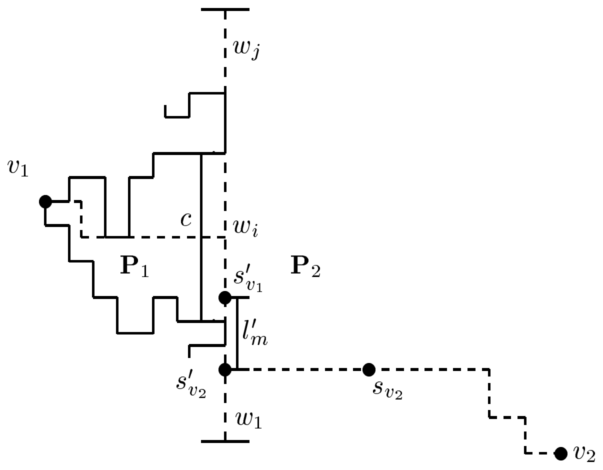

Figure 35: c'_m *may consist of a number of windows.*

that there is a point p above s'_{v_2} that can be reached from v_2 with at most k_2 links (see Figure 36a). Since $c'_m(v_2,k_2)$ is an interval, we can assume that p is below s'_{v_1}. Let $\mathcal{P}$ be a path from v_2 to p of length k_2. Since the rectangle spanned by l_m and l'_m does not contain any points of the boundary of $\mathbf{P}$, we can prolong the last link l of $\mathcal{P}$ so that it reaches c_m. This immediately implies that v_2 can reach points above s_{v_2} on c_m with k_2 links which contradicts the definition of s_{v_2}.

We now assume that v_2 is *not* to the right of c. Then, a shortest path $\mathcal{P}$ from v_2 to c'_m crosses c (see Figure 36b). Since c is above s'_{v_2} this implies that v_2 can reach points on c'_m above s'_{v_2} with k_2 links which contradicts the above observation.

Finally, we show that $d(v_2, c) \geq k_2+1$. First assume that v_2 is to the right of c'_m. Then, a shortest path from v_2 to c intersects c'_m which immediately implies that either v_2 can reach c'_m with less than $d(v_2, c)$ links or v_2 can reach points above s'_{v_2} on c'_m with $d(v_2, c)$ links both of which leads to a contradiction if $d(v_2, c) \leq k_2$.

Hence, assume v_2 is to the left of c'_m (see Figure 37). Let h be the horizontal line segment from the lower end point of c to the right. Clearly, v_2 is below h since otherwise it can reach points above s'_{v_2} with k_2 links. Let $\mathcal{P}$ be a shortest path from v_2 to c. Since v_2 is below h and c is above h, $\mathcal{P}$ crosses h. The first vertical link l that intersects h is at least at distance k_2 from v_2 since otherwise we again have that v_2 can reach points above s'_{v_2} with k_2 links. This implies that $\lambda(\mathcal{P}) \geq k_2 + 1$. □

We can summarize the result of our considerations as follows. We can find a chord c in linear time such that $d(v_1, c) = k_1$, $d(v_2, c) \geq k_2 + 1$, and that v_1 and v_2 are on different sides of c. These are the only properties of c we will make use of in the rest of this section. From now on we have to treat the two cases D is even and D is odd separately. As usual we define $d_1 = \max_{v \in \mathbf{P}_1} d(v, c)$ and $d_2 = \max_{v \in \mathbf{P}_2} d(v, c)$. Furthermore, for simplicity we use k to denote k_1. First we consider the case that D is even.

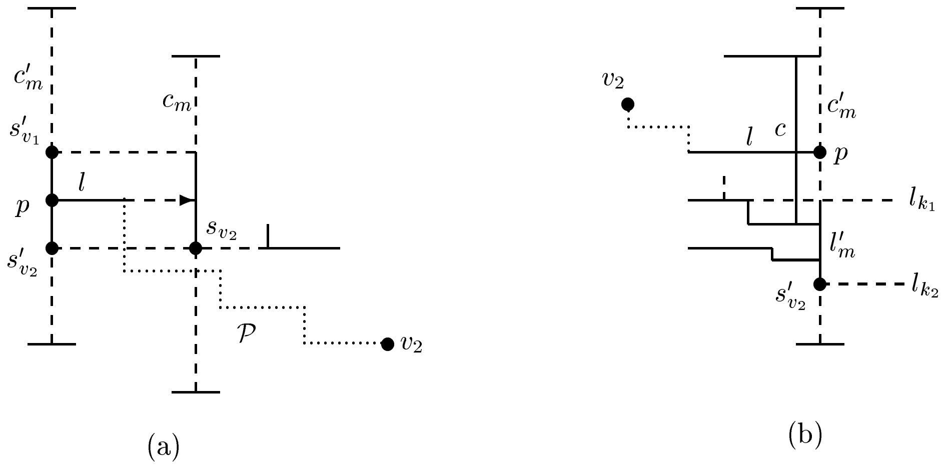

Figure 36: *v_2 cannot reach any point above s'_{v_2}.*

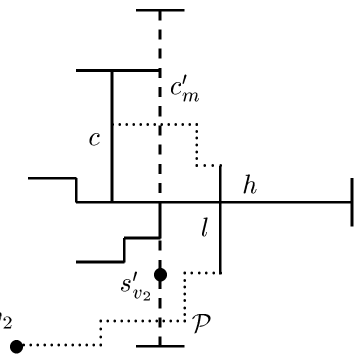

Figure 37: *v_2 needs $k+1$ links to reach c.*

D is Even

If D is even, we have that $D = 2k$ and $k_2 = k-1$. We will show that $R \le k+1 = \lceil D/2 \rceil + 1$ and that c satisfies the conditions of Theorem 5. Note that we have $d(v_1, c) = d(v_2, c) = k$.

Our first observation is that we cannot have vertices $v' \in \mathbf{P}_1$ with $d(v', c) = k + 1$ and vertices $v'' \in \mathbf{P}_2$ with $d(v'', c) = k + 1$ at the same time since they would have a distance $d(v', v'')$ which is at least $(k + 1) + (k + 1) - 1 > D$ by Lemma 2.3. So w.l.o.g. we assume that there is no vertex v in $\mathbf{P}_2$ with $d(v, c) = k + 1$, i.e., $d_2 = k$. We claim that $\bigcap_{v \in \mathbf{P}_1} c(v, k + 1)$ is non-empty. For otherwise there are two vertices v' and v'' in $\mathbf{P}_1$ with $c(v', k+1) \cap c(v'', k+1) = \emptyset$ and by Lemma 3.2 we have that $d(v', v'') \ge (k+1)+(k+1)-1 > D$.

If p is a point in $\bigcap_{v \in \mathbf{P}_1} c(v, k+1)$, p can reach all vertices in $\mathbf{P}_1$ with $k+1$ links and also all the vertices in $\mathbf{P}_2$ since we have $d(v, c) \le k$, for all vertices v in $\mathbf{P}_2$. Hence, $R \le k + 1$. Furthermore, we have that $k + 1 \ge d_1 \ge d(v_1, c) = k$ and $d_2 = k$ and, hence, c satisfies the conditions of Theorem 5.

Next we consider the case that D is odd which turns out to be quite a bit more involved.

D is Odd

If D is odd, then we have that $D = 2k + 1$ and $k_2 = k$. Unfortunately, it may happen that c does not meet the conditions of Theorem 5 since the radius can be as large as $k + 2$; but in this case we can compute other chords which are somewhat in the "neighbourhood" of c that do. Again note that we have $d(v_1, c) = k$ and $d(v_2, c) = k + 1$ by Lemma 6.2.

Our first claim is that $R \le k+2 = \lceil D/2 \rceil + 1$ which completes the proof of Lemma 2.2. To see this we first observe that there can be no vertex v in $\mathbf{P}_1$ with $d(v, c) \ge k + 2$ since this implies that $d(v, v_2) \ge (k + 2) + (k + 1) - 1 > D$ by Lemma 2.3. Furthermore, since $d(v_1, c) = k$, a vertex in $\mathbf{P}_2$ cannot have a distance of $k + 3$ or more links to c by the same argument. This, in particular, implies that a point p on c can reach all the points in $\mathbf{P}_1$ with $k + 2$ links. For if v is a vertex in $\mathbf{P}_1$, then $d(v, c) \le k + 1$ and p can reach $c(v, k + 1)$ with one link. Hence, we have $d(p, v) \le k + 2$.

Next we show that there are points on c that can also reach all vertices in $\mathbf{P}_2$ with $k + 2$ links. From this it immediately follows that $R \le k + 2$. So let v' and v'' be two vertices in $\mathbf{P}_2$. Then, $c(v', k + 2)$ intersects $c(v'', k + 2)$ since otherwise we have that $d(v', v'') \ge (k + 2) + (k + 2) - 1 > D$ by Lemma 3.2. This, in particular, implies that $c' = \bigcap_{v \in \mathbf{P}_2} c(v, k + 2)$ is non-empty and we can choose a point p in c'. By definition p can reach every vertex in $\mathbf{P}_2$ with $k + 2$ links.

The problem we are facing now is that it is true that $R \le k+2$ and that $d_2 \ge k+1$ but d_1 may be less than $k + 1$ since we only have vertex v_1 in $\mathbf{P}_1$ with distance k to c. Recall that we want to find a chord such that $R - d_1 \le 1$ *and* $R - d_2 \le 1$. Unfortunately, it may happen that c violates these conditions.

In the following we will show how to find a chord that meets the conditions of Theorem 5 in the case that D is odd. We do so by an extensive case analysis. Though we do not say so explicitly, we claim in each subcase that we can carry out all the necessary computations

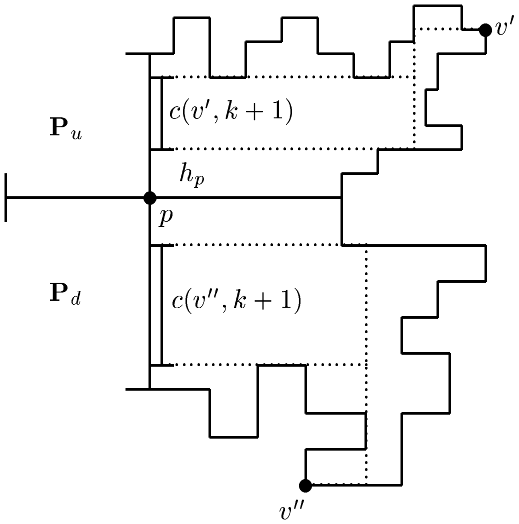

Figure 38: *$c(v', k+1)$ and $c(v'', k+1)$ are disjoint.*

in linear time. Note that we either have to show that there is a chord c' that has vertices at distance $k+1$ on both sides or that $R \leq k+1$ since in this case c also satisfies the conditions of Theorem 5.

Case 1 There is a vertex v in $\mathbf{P}_1$ with $d(v, c) = k+1$.
Then, we have $d_1 = k+1$, $k+2 \geq d_2 \geq k+1$, and $k+2 \geq R \geq k+1$. Hence, c satisfies the conditions of Theorem 5 and we can choose c as a splitting chord.

Case 2 For all vertices $v \in \mathbf{P}_1$, we have $d(v, c) \leq k$, i.e., $d_1 = k$.
This case divides up into two of subcases.

Case 2.1 For all vertices $v \in \mathbf{P}_2$, we have $d(v, c) \leq k+1$.
Let v' and v'' be two different vertices in $\mathbf{P}_2$ with $d(v', c) \leq k+1$ and $d(v'', c) \leq k+1$. We first assume that $c(v', k+1)$ does not intersect $c(v'', k+1)$ (for illustration see Figure 38). Hence, we can choose a point p on c between $c(v', k+1)$ and $c(v'', k+1)$. Let h_p be the horizontal chord through p. We claim that neither v' nor v'' can reach h_p with k links. For if $d(v', h_p) \leq k$, then v' can reach p with an admissible path of $k+1$ links although we chose p not to be in $c(v', k+1)$. For further reference we state this as a lemma.

Lemma 6.3 *Let c be a chord in $\mathbf{P}$, v a vertex in $\mathbf{P}$, and d_v an integer such that $c(v, d_v)$ is non-empty. If p is a point on c not in $c(v, d_v)$, then v cannot reach the chord c_p orthogonal to c through p with $d_v - 1$ links.*

Note that the above lemma holds even if $d(v, c) < d_v$. The only assumption we make is that p is not contained in $c(v, d_v)$, i.e., that $c(v, d_v) \neq c$.

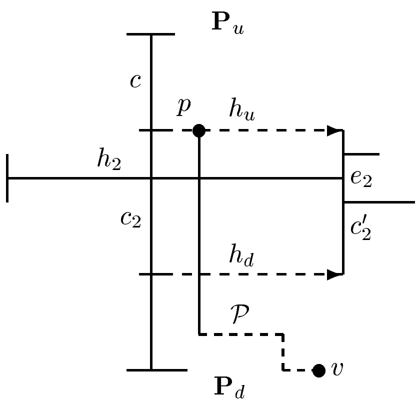

Figure 39: *The definition of h_2.*

Let $\mathbf{P}_u$ be the part of $\mathbf{P}$ above h_p and $\mathbf{P}_d$ the part below it. We have that $\max_{v \in \mathbf{P}_u} d(v, h_p) \geq k+1$ and $\max_{v \in \mathbf{P}_d} d(v, h_p) \geq k+1$. Hence, h_p satisfies the conditions of Theorem 5.

So now suppose that the subchords $c(v', k+1)$ and $c(v'', k+1)$ of any two vertices v' and v'' in $\mathbf{P}_2$ intersect. This implies that $c_1 = \bigcap_{v \in \mathbf{P}_2} c(v, k+1)$ is non-empty. If we choose a point p in c_1, then this point can reach all the points in $\mathbf{P}_2$ with $k+1$ links and also all the points in $\mathbf{P}_1$ since $d_1 = k$. Thus, $R = k+1$ and we can use c as a splitting chord in Theorem 5.

Case 2.2 There is a vertex v in $\mathbf{P}_2$ with $d(v, c) = k+2$.

Let v' and v'' be two arbitrary vertices in $\mathbf{P}_2$. As before it can be shown that all pairs of $(k+2)$-subsegments intersect and, thus, $c_2 = \bigcap_{v \in \mathbf{P}_2} c(v, k+2)$ is non-empty. Note that all the intervals $c(v', k)$ with v' in $\mathbf{P}_1$ intersect c_2 since otherwise there is a vertex v'' in $\mathbf{P}_2$ with $c(v', k) \cap c(v'', k+2) = \emptyset$ and, hence, $d(v', v'') \geq k + (k+2) > D$ by Lemma 2.3. We move the subedge c_2 of c to the right until we hit an edge e_2 on the boundary of $\mathbf{P}$. This is illustrated in Figure 39. Let the new line segment be c_2'. We denote the horizontal chord through the lower end points of c_2 and c_2' by h_d and the horizontal chord through the upper end points by h_u. Furthermore, let h_2 be a horizontal chord in $\mathbf{P}$ from some point on $e_2 \cap c_2'$ towards the left. We show that in some cases h_2 can be used as a splitting chord.

First we claim that all vertices in $\mathbf{P}$ can reach h_2 with an admissible path of length $k+1$. To see this let $\mathbf{P}_u$ be the part of $\mathbf{P}$ above h_2 and $\mathbf{P}_d$ the part below it. Let v be a vertex in $\mathbf{P}_2$. We want to show that $d(v, h_2) \leq k+1$. W.l.o.g. assume that v belongs to $\mathbf{P}_d$. v can reach some point p on h_u with a path $\mathcal{P}$ of length $k+1$ links since v can reach the upper end point of c_2 with an admissible path of length $k+2$. $\mathcal{P}$ crosses h_2 since v is below h_2 and p is above it. This immediately implies that $d(v, h_2) \leq k+1$. Clearly, all vertices in $\mathbf{P}_1$ can reach h_2 with $k+1$ links since c intersects h_2 and we assume that $\max_{v \in \mathbf{P}_1} d(v, c) = k$. Therefore, all vertices in $\mathbf{P}$ can reach h_2 with $k+1$ links. Note that we cannot have that, for all vertices v in $\mathbf{P}_2$, $d(v, h_2) \leq k$ since we assume that there is a vertex v_2 with $d(v_2, c) = k + 2$.

So, there is a vertex v in $\mathbf{P}_2$ with $d(v, h_2) \geq k + 1$. If there are vertices at distance $k + 1$ in both $\mathbf{P}_u$ and $\mathbf{P}_d$, then h_2 meets the conditions of Theorem 5 and we are done.

Therefore, w.l.o.g. we assume that there are only points in $\mathbf{P}_u$ that have distance $k + 1$ to h_2, i.e., we have that $\max_{v \in \mathbf{P}_d} d(v, h_2) \leq k$. If $h_2' = \bigcap_{v \in \mathbf{P}_u} h_2(v, k + 1)$ is non-empty, then a point p in h_2' can reach all the vertices in $\mathbf{P}_u$ with $k + 1$ links. Furthermore, p can also reach all the vertices in $\mathbf{P}_d$ with $k + 1$ links since we have $d(v, h_2) \leq k$, for all $v \in \mathbf{P}_d$. This, in particular, implies that $R \leq k + 1$ and, hence, c can be chosen as a splitting chord in Theorem 5.

Finally, if h_2' is empty, there are two vertices in $\mathbf{P}_u$ that have disjoint $k + 1$ intervals. Lemma 6.3—together with the same argument as in Case 2.1—yields that there is a vertical chord c' orthogonal to h_2 that has one point with distance $k + 1$ on each side. Hence, c' satisfies the conditions of Theorem 5.

We have shown that by choosing either c, h_p, h_2, or c' as a splitting chord we obtain a chord that satisfies the conditions of Theorem 4. Clearly, we can compute all the chords in linear time and also check the conditions under which to choose either of them. This completes the algorithm.

7 The Number of Components of the Link Center

In this section we prove that the rectilinear link center consists of at most two components. In order to do so we employ again the technique used previously. Our first observation is that no chord can intersect two components of the link center of $\mathbf{P}$.

LEMMA 7.1 *If $\mathbf{C}_1$, $\mathbf{C}_2$ are two components of $lc(\mathbf{P})$ and c is a rectilinear chord in $\mathbf{P}$, then c cannot intersect both $\mathbf{C}_1$ and $\mathbf{C}_2$.*

PROOF: Let $\mathbf{C}_1$, $\mathbf{C}_2$ be two components of $lc(\mathbf{P})$ and c a chord in $\mathbf{P}$. We prove the claim by contradiction. Suppose that c intersects $\mathbf{C}_1$ and $\mathbf{C}_2$ (see Figure 40). Let $c_1 = c \cap (\mathbf{C}_1 \cup \mathbf{C}_2)$. c_1 is contained in the intervals $c(v, R)$, for all $v \in \mathbf{P}$, since we can reach every point in $\mathbf{P}$ with R links from $\mathbf{C}_1$ and $\mathbf{C}_2$. Hence, we have that $c_1 \subseteq \bigcap_{p \in \mathbf{P}} c(p, R)$. Note that $\bigcap_{p \in \mathbf{P}} c(p, R)$ is again an interval. Therefore, $\mathbf{C}_1$ and $\mathbf{C}_2$ are connected by the points in $\bigcap_{p \in \mathbf{P}} c(p, R)$ which contradicts the fact that $\mathbf{C}_1$ and $\mathbf{C}_2$ are two components of the link center of $\mathbf{P}$. □

So let $\mathbf{C}_1$ and $\mathbf{C}_2$ be two components of $lc(\mathbf{P})$. We claim that there is a vertical chord c such that $\mathbf{C}_1$ is on one side of c and $\mathbf{C}_2$ on the other. To see this we first choose a vertical chord c' that intersects one component, say $\mathbf{C}_1$. c' splits $\mathbf{P}$ into two components $\mathbf{P}_1'$ to the left of it and $\mathbf{P}_2'$ to the right of it. W.l.o.g. assume that $\mathbf{C}_2$ is contained in $\mathbf{P}_2'$. We now move c' to the right until it does not intersect $\mathbf{C}_1$ any more. It might happen that c' sweeps over a "cliff" and $\mathbf{C}_2$ is then to the left of c'. In this case we move c' into the window that cuts off the part of $\mathbf{P}$ that contains $\mathbf{C}_2$ (see Figure 41). Since $\mathbf{C}_1$ and $\mathbf{C}_2$ are closed sets, there is a position such that c' does not intersect $\mathbf{C}_1$ and $\mathbf{C}_2$ nor any other component of

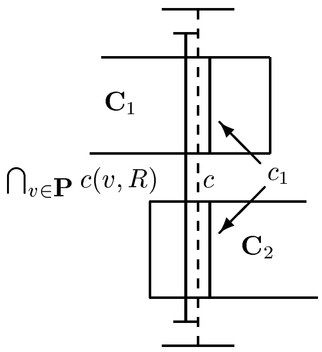

Figure 40: *c intersects $\mathbf{C}_1$ and $\mathbf{C}_2$.*

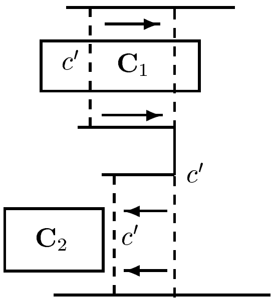

Figure 41: *c' sweeps over a "cliff".*

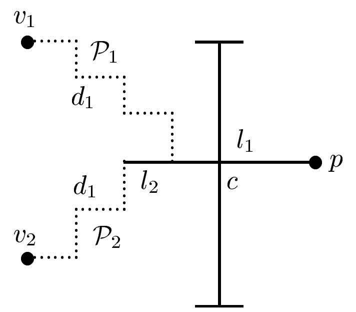

Figure 42: *If d_1 equals R, then p belongs to the maximal histogram of I_1.*

$lc(\mathbf{P})$. So let c be a vertical chord such that $\mathbf{C}_1$ is to the left of c and $\mathbf{C}_2$ is to the right of it. In the following we denote as usual the part of $\mathbf{P}$ to the left of c by $\mathbf{P}_1$ and the part to the right of c by $\mathbf{P}_2$. Furthermore, we define $d_i = \max_{v \in \mathbf{P}_i} d(v, c)$, for $i = 1, 2$. Before we continue with our investigations we need the following technical lemma. As before we define $\mathcal{I}_1$ as the set of intervals $\{c(v, d_1) \mid v \in \mathbf{P}_1\}$ on c and I_1 as the intersection of all intervals in $\mathcal{I}_1$, i.e., $I_1 = \bigcap \mathcal{I}_1$. Recall that $\mathcal{I}_1$ can be computed in linear time by [B89]. Further, we denote the maximal histogram of I_1 in $\mathbf{P}_2$ by $\mathbf{H}_1$.

LEMMA 7.2 *If $d_1 = R$, then*

1. *if $I_1 \neq \emptyset$, then $lc(\mathbf{P}) \cap \mathbf{P}_2$ is contained in $\mathbf{H}_1$.*

2. *if $I_1 = \emptyset$, then no point in $\mathbf{P}_2$ is in the link center of $\mathbf{P}$.*

PROOF: Let p be a point in the part of the link center in $\mathbf{P}_2$. We show that I_1 is non-empty and that p is contained in $\mathbf{H}_1$ in this case. This then proves both claims. Note that we can choose two furthest neighbours v_1 and v_2 of c in $\mathbf{P}_1$ such that $c(v_1, d_1) \cap c(v_2, d_1) = I_1$. Let $\mathcal{P}_1$ be a shortest path from p to v_1 and $\mathcal{P}_2$ a shortest path from p to v_2. Furthermore, let l_1 be the first horizontal link of $\mathcal{P}_1$ as seen from v_1 that intersects c (see Figure 42). Since $d(v_1, c) = d_1$, l_1 is the ${d_1}^{\text{st}}$ link from v_1 on $\mathcal{P}_1$. We also have that the first horizontal link l_2 of $\mathcal{P}_2$ that intersects c is the ${d_1}^{\text{st}}$ link from v_2.

Since $d_1 = R$ and p is contained in the link center, p is the end point of l_1 and l_2. Hence, l_1 and l_2 intersect the same point of c and $c(v_1, d_1) \cap c(v_2, d_1) = I_1$ is non-empty. Furthermore, p is contained in the maximal histogram of I_1. $\square$

The first obversation about a chord that has components on both sides is that d_1 and d_2 equal R.

LEMMA 7.3 *If c is a chord outside the link center of $\mathbf{P}$ such that $\mathbf{C}_1$ and $\mathbf{C}_2$ are on different sides of c, then $R = d_1 = d_2$.*

PROOF: W.l.o.g. we assume that $d_1 \geq d_2$. If $d_1 > R$, then there can be no point of the link center in $\mathbf{P}_2$ as we observed in Section 5. Hence, we have that $R \geq d_1 \geq d_2$. To see that we also have $d_2 \geq R$ suppose the contrary, i.e., that $R > d_2$. If we have $R > d_1$ as well, then $c \subseteq lc(\mathbf{P})$ since any point on c can reach all points in $\mathbf{P}$ with $\max(d_1, d_2) + 1 \leq R$ links which contradicts our choice of c. Hence, we can assume that $R = d_1$. Lemma 7.2 implies that $I_1 \neq \emptyset$ since $\mathbf{C}_2$ is a non-empty part of the link center in $\mathbf{P}_2$. If we take a point p in I_1, then p can reach all vertices in $\mathbf{P}_1$ with R links by the definition of I_1 and also all vertices in $\mathbf{P}_2$ since $d_2 + 1 \leq R$. Hence, p belongs to the link center again in contradiction to our choice of c. Hence, we have that $d_2 \geq R$ and we have proved that $R \geq d_1 \geq d_2 \geq R$ which implies equality. □

We define $I_2 = \bigcap_{v \in \mathbf{P}_2} c(v, d_2)$. Since $d_1 = d_2 = R$, we immediately have that I_1 and I_2 are non-empty by Lemma 7.2. Furthermore, Lemma 7.2 implies that the part of the link center in $\mathbf{P}_1$ and $\mathbf{P}_2$ is contained in the two maximal histograms $\mathbf{H}_1 \subseteq \mathbf{P}_2$ of I_1 and $\mathbf{H}_2 \subseteq \mathbf{P}_1$ of I_2. W.l.o.g. we assume in the following that I_1 is above I_2.

We prove the claim that the link center of $\mathbf{P}$ consists of at most two components by contradiction. So assume that the link center of $\mathbf{P}$ consists of more than two components and let $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{C}_3$ be any three of them. Furthermore, we choose a chord c such that $\mathbf{C}_1 \subseteq \mathbf{H}_2$ and $\mathbf{C}_2 \subseteq \mathbf{H}_1$ with $\mathbf{H}_1$ and $\mathbf{H}_2$ as defined above. In the following we will show that the existence of $\mathbf{C}_3$ implies that there are points on c either below I_1 that can reach all vertices in $\mathbf{P}_1$ with R links or above I_2 that can reach all vertices in $\mathbf{P}_2$ with R links.

As the above argument shows $\mathbf{C}_3$ is contained in one of $\mathbf{H}_1$ or $\mathbf{H}_2$. W.l.o.g. we assume that $\mathbf{C}_3$ is contained in the maximal histogram $\mathbf{H}_1$ of I_1 in $\mathbf{P}_2$. As before we can find a vertical chord c' that separates $\mathbf{C}_2$ and $\mathbf{C}_3$. Suppose that $\mathbf{C}_3$ is to the right of c' and $\mathbf{C}_2$ to the left of it. The situation is illustrated in Figure 43.

c' splits $\mathbf{P}$ into two components $\mathbf{P}'_1$ and $\mathbf{P}'_2$ to the left and right of c'. Let I'_1 and I'_2 on c' be defined analogously to I_1 and I_2 and let $\mathbf{H}'_2$ be the maximal histogram of I'_2 in $\mathbf{P}'_1$. Furthermore, we denote the maximal distance of a vertex in $\mathbf{P}'_1$ to c' by d'_1 and maximal distance of a vertex in $\mathbf{P}'_2$ to c' by d'_2.

Since $\mathbf{P}_1 \subseteq \mathbf{P}'_1$, we have $\mathbf{C}_1 \subseteq \mathbf{H}_2 \cap \mathbf{H}'_2$. Note that is only true if $d_1 = d_2$ which holds by Lemma 7.3. Hence, the y-ranges of I_2 and I'_2 overlap. Since I_2 is below I_1 whose y-range intersects the y-ranges of I'_1 and I'_2, I'_1 is above I'_2 and I_2 is below I'_1.

Now let h_1 be the horizontal line segment from the upper end point of I'_1 to c and h_2 the horizontal line segment from the lower end point of I'_2 to c. Let p_1 be the right end point of h_1 and p_2 be the left end point of h_2. We claim that h_1 and h_2 are contained in $\mathbf{P}$. For assume that a vertical edge of $\mathbf{P}$ intersects h_1. Then, no shortest path from a point in $\mathbf{P}_1$ to p_1 intersects c with the last link and, hence, $d_1 < d'_1 = R$ which contradicts Lemma 7.3. Similarly, we have that if a vertical edge of $\mathbf{P}$ intersects h_2, then no path from a point in $\mathbf{P}'_2$ to p_2 intersects c' with the last link and $d'_2 < d_2 = R$. Hence, the rectangle spanned by h_1 and h_2 is empty and we can prolong all admissible paths from vertices in $\mathbf{P}'_2$ to I'_2 to c. This, in particular, implies that all vertices in $\mathbf{P}'_2$ can reach the projection I of I'_2 onto c with R links.

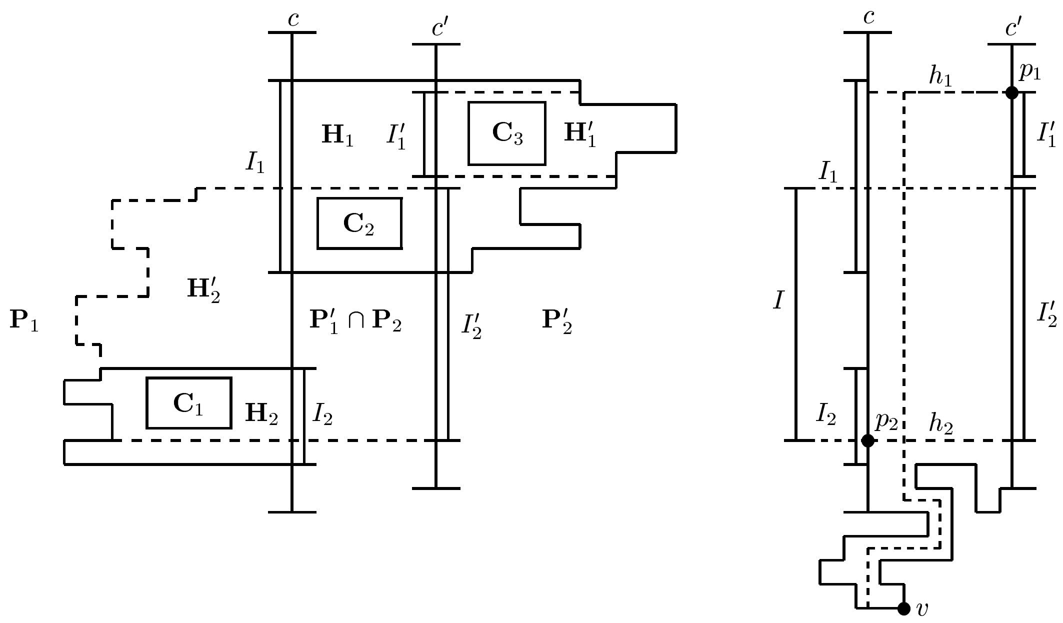

Figure 43: *The link center cannot consist of three components.*

If we can show that this is also true for the vertices in $\mathbf{P}_1' \cap \mathbf{P}_2$, i.e., the vertices between c and c', then I can be reached from all vertices in $\mathbf{P}_2$ with R links which contradicts the fact that I contains points that are above I_2. So let v be a vertex in $\mathbf{P}_1' \cap \mathbf{P}_2$. Since the rectangle spanned by h_1 and h_2 is empty, v is either above h_1 or below h_2. If v is below h_2, then it reaches p_1 with R links and either intersects c or c' with the $(R-2)^{\text{nd}}$ link or h_2 with the $(R-1)^{\text{st}}$ link. In each case we can reach all the points above h_2 on c with R links. If v is above h_1, the same argument shows that v reaches all points on I with R links.

Hence, all vertices in $\mathbf{P}_2$ can reach I with R links which contradicts the fact that I_1 and I_2 are disjoint. Therefore, there can be only two components in the link center of $\mathbf{P}$. We have shown the following theorem.

Theorem 6 *If* $\mathbf{P}$ *is a simple rectilinear polygon, then the rectilinear link center of* $\mathbf{P}$ *consists of at most two components.*

References

[B89] Mark de Berg. *On Rectilinear Link Distance.* Technical Report RUU-CS-89-13, Department of Computer Science, University of Utrecht, P.O.Box 80.089, 3502 TB Utrecht, the Netherlands, May 1989.

[BKNO90] M.T. de Berg, M.J. van Kreveld, B.J. Nilsson, M.H. Overmars. Finding Shortest Paths in the Presence of Orthogonal Obstacles Using a Combined L_1 and Link Metric. In *Proc. 2nd Scandinavian Workshop on Algorithm Theory*, pages 213–224, 1990.

[Cha90] Bernard Chazelle. Triangulating a Simple Polygon in Linear Time. In *Proc. 31th Symposium on Foundations of Computer Science*, pages 220–230, 1990.

[DLS89] H.N. Djidjev, A. Lingas, J.-R. Sack. An $O(n \log n)$ Algorithm for Computing a Link Center in a Simple Polygon. In *Proc. STACS*, pages 96–107, 1989.

[Ke89] Yan Ke. An Efficient Algorithm for Link-distance Problems. In *Proceedings of the Fifth Annual Symposium on Computational Geometry*, pages 69–78, ACM, ACM Press, Saarbrücken, West Germany, June 1989.

[LPS*87] W. Lenhart, R. Pollack, J.-R. Sack, R. Seidel, M. Sharir, S. Suri, G. Toussaint, S. Whitesides, C. Yap. Computing the Link Center of a Simple Polygon. In *Proceedings of the Third Annual Symposium Computational Geometry*, pages 1–10, ACM, ACM Press, Waterloo, Ontario, Canada, June 1987.

[Lev87] Christos Levcopoulos. *Heuristics for Minimum Decompositions of Polygons.* PhD thesis, University of Linköping, Linköping, Sweden, 1987.

[Sur87] Subhash Suri. *Minimum Link Paths in Polygons and Related Problems.* PhD thesis, Johns Hopkins University, Baltimore, Maryland, August 1987. pages 213–224, 1990.