Rectilinear Link Diameter and Radius in a Rectilinear Polygonal Domain[*] [†]

Man-Kwun Chiu[1,2], Elena Khramtcova[3], Aleksandar Markovic[4], Yoshio Okamoto[5,6], Aurélien Ooms[3], André van Renssen[1,2], and Marcel Roeloffzen[1,2]

1 **National Institute of Informatics, Tokyo, Japan**
`{chiumk,andre,marcel}@nii.ac.jp`
2 **JST, ERATO, Kawarabayashi Large Graph Project**
3 **Université libre de Bruxelles (ULB), Brussels, Belgium**
`elena.khramtsova@gmail.com, aureooms@ulb.ac.be`
4 **TU Eindhoven, Eindhoven, the Netherlands**
`a.markovic@tue.nl`
5 **University of Electro-Communications, Tokyo, Japan**
`okamotoy@uec.ac.jp`
6 **RIKEN Center for Advanced Intelligence Project, Tokyo, Japan**

Abstract

We study the computation of the diameter and radius under the rectilinear link distance within a rectilinear polygonal domain of n vertices and h holes. We introduce a *graph of oriented distances* to encode the distance between pairs of points of the domain. This helps us transform the problem so that we can search through the candidates more efficiently. Our algorithm computes both the diameter and the radius in $O(\min(n^\omega, n^2 + nh \log h + \chi^2))$ time, where $\omega < 2.373$ denotes the matrix multiplication exponent and $\chi \in \Omega(n) \cap O(n^2)$ is the number of edges of the graph of oriented distances. We also provide a faster algorithm for computing the diameter that runs in $O(n^2 \log n)$ time.

1 Introduction

Diameters and radii are popular characteristics of metric spaces. For a compact set S with a metric $d\colon S \times S \to \mathbb{R}^+$, its diameter is defined as $\operatorname{diam}(S) := \max_{p \in S} \max_{q \in S} d(p, q)$, and its radius is defined as $\operatorname{rad}(S) := \min_{p \in S} \max_{q \in S} d(p, q)$. The points that realize these distances are called the *diametral pair* and *center*, respectively. All of these terms are the natural extensions of the same concepts in a disk and give some interesting properties of the environment, such as the worst-case response time or ideal location of a serving facility.

Much research has been devoted towards finding efficient algorithms to compute the diameter and radius for various types of sets and metrics. In computational geometry, one of the most well-studied and natural metric spaces is a polygon in the plane. This paper focuses on the computation of the diameter and the radius of a rectilinear polygon, possibly with holes (i.e., a *rectilinear polygonal domain*) under the *rectilinear link distance*. Intuitively,

[*] MC, AvR and MR were supported by JST ERATO Grant Number JPMJER1201, Japan. EK was partially supported by the SNF Early Postdoc Mobility grant P2TIP2-168563, Switzerland, and F.R.S.-FNRS, Belgium. AM was supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 024.002.003. YO was partially supported by JSPS KAKENHI Grant Number 15K00009 and JST CREST Grant Number JPMJCR1402, and Kayamori Foundation of Informational Science Advancement. AO was supported by the Fund for Research Training in Industry and Agriculture (FRIA).

[†] Omitted proofs and further details can be found in the full version [4]

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

this metric measures the minimum number of links (segments) required in any rectilinear path connecting two points in the domain, where rectilinear indicates that we are restricted to horizontal and vertical segments only.

Many problems that are easy under the L_1 or Euclidean metric turn out to be more challenging under the link distance. For example, computing the shortest path between two points in a polygonal domain can be done in $O(n \log n)$ time for both Euclidean [8] and L_1 metrics [10, 11]. However, even approximating the same within a factor of $(2 - \epsilon)$ under the link distance is 3-SUM hard [12], and thus it is unlikely that a significantly subquadratic-time algorithm is possible.

Computing the diameter and radius is no exception: when considering simple polygons (i.e., polygons without holes) of n vertices, the diameter and center can be found in linear time for both Euclidean [1, 7] and L_1 metrics [2]. However, the best known algorithms for the link distance run in $O(n \log n)$ time [5, 16]. Lowering the running times or proving the impossibility of this is a longstanding open problem in the field. The only partial answer to this question was given by Nilsson and Schuierer [14, 15]; they showed that the diameter and center can be found in linear time when we are only allowed to use rectilinear paths.

If we consider polygons with holes, the difference becomes even bigger: no algorithm for computing the diameter and radius under the link distance is known, not even one that runs in exponential time. In comparison, polynomial-time algorithms are known both for diameter and radius under L_1 and Euclidean metrics.

1.1 Results

We introduce the *graph of oriented distances*, a graph that implicitly encodes the distance between regions of the domain. In Section 3 we use this graph to transform the problem: rather than searching pairwise distances in a list of potential candidates for diameter or center, we transform the problem into a rectangle intersection problem. Intuitively speaking, we cover the domain with several rectangles, and we find two pairs of rectangles that pairwise intersect (and satisfy other properties). In particular, once we have found the diametral pair, the four rectangles that satisfy the property can be used as a *witness*.

This transformation leads to an algorithm for computing both the rectilinear link diameter and radius of a rectilinear polygonal domain with n vertices and h holes. The algorithm is described in Section 4 and runs in $O(n^\omega)$ time, where $\omega < 2.373$ is the matrix multiplication exponent [9]. Alternatively, we can also bound the running time in terms of the number χ of edges of the graph of oriented distances (χ will range from $\Omega(n)$ to $O(n^2)$ depending on P). With this parameter the running time becomes $O(n^2 + nh \log h + \chi^2)$. In Section 5 we use a different approach to obtain an $O(n^2 \log n)$ time algorithm to compute the diameter. All of the algorithms presented in the paper can be modified to return not only diameter or radius, but also the points that realize it (i.e., diametral pair and center).

1.2 Preliminaries

A *rectilinear simple polygon* (also called an orthogonal polygon) is a simple polygon that has horizontal and vertical edges only. A *rectilinear polygonal domain* P with h pairwise disjoint holes and n vertices is a connected and compact subset of $\mathbb{R}^2$ with h pairwise disjoint holes, in which the boundary of each hole is a simple closed rectilinear curve.

A *rectilinear path* π from $p \in P$ to $q \in P$ is a path from p to q that consists of vertical and horizontal segments, each contained in P, and such that along π each vertical segment is followed by a horizontal one and vice versa. Recall that P is a closed set, so π can traverse the boundary of P (along the outer face and any of the h obstacles). We define the *link length* of such a path to be the number of segments composing it. The *rectilinear link distance* between points $p, q \in P$ is defined as the minimum link length of a rectilinear path from p to q, and denoted by $\ell_P(p, q)$. It is well known that in rectilinear polygonal domains there always exists a rectilinear polygonal path between any two points $p, q \in P$, and thus the distance is well defined. Once the distance is defined, the definitions of *rectilinear link diameter* $\mathrm{diam}(P)$ and *rectilinear link radius* $\mathrm{rad}(P)$ directly follow. For simplicity in the description, we assume that a pair of vertices do not share the same x- or y-coordinate unless they are connected by an edge.

2 Graph of Oriented Distances

For any domain P, we virtually shoot a ray left and right from any horizontal segment of the domain until it hits another segment of P, partitioning it into rectangles. We call this partition the *horizontal decomposition*, $\mathcal{H}(P)$. Similarly, if we shoot rays up and down from vertical segments, we get the *vertical decomposition*, $\mathcal{V}(P)$. Observe that both decompositions have linear size and can be computed in $O(n \log n)$ time with a plane sweep.

Given two rectangles $i, j \in \mathcal{H}(P) \cup \mathcal{V}(P)$, we use $i \sqcap j$ to denote the boolean operation which returns *true* if and only if (1) the rectangles i and j properly intersect (i.e. their intersection has non-zero area), and (2) one of i, j belongs to $\mathcal{H}(P)$, and the other to $\mathcal{V}(P)$.

▶ **Definition 2.1** (Graph of Oriented Distances)**.** Given a rectilinear polygonal domain P we define the undirected graph $\mathcal{G}(P) = (\mathcal{H}(P) \cup \mathcal{V}(P), \{\, (h, v) \in \mathcal{H}(P) \times \mathcal{V}(P) \colon\ h \sqcap v \,\})$.

In other words, vertices of $\mathcal{G}(P)$ correspond to rectangles of the horizontal and the vertical decompositions of P. We add an edge between two vertices if and only if the corresponding rectangles properly intersect. Note that this graph is bipartite, and has $O(n)$ vertices. From now on, we make a slight abuse of notation and identify a rectangle with its corresponding vertex (thus, we talk about the neighbors of a rectangle $i \in \mathcal{H}(P)$ in $\mathcal{G}(P)$, for example).

The name *Graph of Oriented Distances* is easily explained: consider a rectilinear path π between two points in P. Each horizontal edge of π is contained in a rectangle of $\mathcal{H}(P)$ and each vertical edge is contained in a rectangle of $\mathcal{V}(P)$. A bend in the path takes place in the intersection of the rectangles containing the two adjacent edges and corresponds to an edge of $\mathcal{G}(P)$. So every rectilinear path π has a corresponding path π' in $\mathcal{G}(P)$ and *vice versa*. Moreover, each bend of π is associated with an edge of π'.

▶ **Definition 2.2** (Oriented distance)**.** Given a rectilinear polygonal domain P, let i and j be two vertices of $\mathcal{G}(P)$, let $\Delta(i, j)$ to be the length of the shortest path between i and j in graph $\mathcal{G}(P)$ plus one. We also define $\Delta(i, i) = 1$.

We first list some useful properties of the oriented distance and then show the relationship between the oriented distance $\Delta(\cdot, \cdot)$ in $\mathcal{G}(P)$ and the link distance $\ell_P(\cdot, \cdot)$ in P.

▶ **Lemma 2.3.** *Let i, j, i', j' be any (not necessarily distinct) rectangles in $\mathcal{H}(P) \cup \mathcal{V}(P)$ such that $i \sqcap i'$, and $j \sqcap j'$. Then, the following hold: (a) $\Delta(i, j) = \Delta(j, i)$, (b) $\Delta(i', j) \in \{\, \Delta(i, j) - 1, \Delta(i, j) + 1 \,\}$, and (c) $\Delta(i', j') \in \{\, \Delta(i, j) - 2, \Delta(i, j), \Delta(i, j) + 2 \,\}$.*

▶ **Lemma 2.4.** *Let p and q be two points of the rectilinear polygonal domain P. If p and q lie in the same vertical or horizontal rectangle of $\mathcal{V}(P)$ or $\mathcal{H}(P)$ then $\ell_P(p, q) = 1$ (if p and q share a coordinate) or $\ell_P(p, q) = 2$ (if both x- and y-coordinates of p and q are distinct). Otherwise, let $i \in \mathcal{H}(P)$, $i' \in \mathcal{V}(P)$, $j \in \mathcal{H}(P)$ and $j' \in \mathcal{V}(P)$ be vertices of the graph of oriented distances such that $p \in i \cap i'$ and $q \in j \cap j'$. Then $\ell_P(p, q) = \min\{\, \Delta(i, j), \Delta(i, j'), \Delta(i', j), \Delta(i', j') \,\}$.*

Intuitively speaking, if we are given two disjoint rectangles $i, j \in \mathcal{H}(P)$, then $\Delta(i, j)$ denotes the minimum number of links needed to connect any two points $p \in i$ and $q \in j$ under the constraint that the first and the last segments of the path are horizontal. It follows that the link distance is the minimum of the four possible options. These $O(n^2)$ distances can be precomputed using algorithms by Mitchell *et al.* [13] in $O(n^2 + nh \log h)$ time or by Chan and Skrepetos [3] in $O(n^2 \log \log n)$.

3 Characterization via Boolean Formulas

Let $\hat{d} = \max_{i,j \in \mathcal{H}(P) \cup \mathcal{V}(P)} \Delta(i, j)$ be the largest distance between vertices of $\mathcal{G}(P)$. Similarly, we define $\hat{r} = \min_{i \in \mathcal{H}(P) \cup \mathcal{V}(P)} \max_{j \in \mathcal{H}(P) \cup \mathcal{V}(P)} \Delta(i, j)$. Note that these two values are the diameter and the radius of $\mathcal{G}(P)$ plus one. We use $\hat{d}$ and $\hat{r}$ to approximate the diameter $\mathrm{diam}(P)$ and radius $\mathrm{rad}(P)$ of a domain P under the rectilinear link distance. First, we relate the distance between two points $p, q \in P$ to the oriented distances between the rectangles that contain p and q. Specifically, from Lemma 2.4, we know that $\ell_P(p, q) = \min\{\, \Delta(i, j), \Delta(i, j'), \Delta(i', j), \Delta(i', j') \,\}$, where $i, j \in \mathcal{H}(P)$ are the horizontal rectangles containing p and q, respectively, and $i', j' \in \mathcal{V}(P)$ are the vertical rectangles containing p and q. Similarly, we define $\hat{\ell}(p, q) = \max\{\, \Delta(i, j), \Delta(i, j'), \Delta(i', j), \Delta(i', j') \,\}$.

▶ **Lemma 3.1.** *For any two points $p, q \in P$, let $i, j \in \mathcal{H}(P)$ and $i', j' \in \mathcal{V}(P)$ be the rectangles containing p and q, i.e., $p \in i \cap i'$ and $q \in j \cap j'$. Then, it holds that $\hat{\ell}(p, q) - 2 \le \ell_P(p, q) \le \hat{\ell}(p, q) - 1$.*

This relation allows us to express the rectilinear link diameter of a domain in terms of $\hat{d}$ and the radius in terms of $\hat{r}$.

▶ **Theorem 3.2.** *The rectilinear link diameter* $\mathrm{diam}(P)$ *of a rectilinear polygonal domain P satisfies* $\mathrm{diam}(P) = \hat{d} - 1$ *if and only if there exist $i, i', j, j' \in \mathcal{H}(P) \cup \mathcal{V}(P)$ with $i \sqcap i'$ and $j \sqcap j'$, such that $\Delta(i, j) = \hat{d}$ and $\Delta(i', j') = \hat{d}$. Otherwise,* $\mathrm{diam}(P) = \hat{d} - 2$.

Proof. First observe that for any pair of points $p, q \in P$ we have $\ell_P(p, q) \le \hat{\ell}(p, q) - 1 \le \hat{d} - 1$ by Lemma 3.1. Hence, the diameter of P is at most $\hat{d} - 1$. Similarly, by the definitions of $\hat{d}$ and $\hat{\ell}(\cdot, \cdot)$, there must be a pair of points $p, q \in P$ so that $\hat{\ell}(p, q) = \hat{d}$. Again by Lemma 3.1 it follows that $\mathrm{diam}(P) \ge \ell_P(p, q) \ge \hat{\ell}(p, q) - 2 = \hat{d} - 2$.

Next we show that the diameter is $\hat{d} - 1$ if and only if the above condition holds. If $\Delta(i, j) = \hat{d}$ and $\Delta(i', j') = \hat{d}$, then by Lemma 2.3 and the fact that neither $\Delta(i, j')$ nor $\Delta(i', j)$ can be larger than $\hat{d}$, we know that $\Delta(i, j') = \Delta(i', j) = \hat{d} - 1$. This implies that a pair of points $p \in i \cap i'$ and $q \in j \cap j'$ have $\ell_P(p, q) = \hat{d} - 1$. Thus, the diameter is $\hat{d} - 1$.

Now consider any pair p, q and the set of rectangles $i, j \in \mathcal{H}(P)$ and $i', j' \in \mathcal{V}(P)$ with $p \in i \cap i'$ and $q \in j \cap j'$. Recall that $\ell_P(p, q) = \min\{\Delta(i, j), \Delta(i, j'), \Delta(j', i), \Delta(i', j')\}$. By Lemma 2.3, $\Delta(i, j)$ and $\Delta(i', j')$ must differ by exactly one from $\Delta(i', j)$ and $\Delta(i, j')$. That implies that two distances may be $\hat{d} - 1$, but if the condition in the lemma is not satisfied, at most one can be $\hat{d}$ and the fourth must be $\hat{d} - 2$ or less. Therefore, if the condition is not satisfied for i, i', j, j', then the diameter is indeed $\hat{d} - 2$. ◀

▶ **Theorem 3.3.** *The rectilinear link radius* $\mathrm{rad}(P)$ *of a rectilinear polygonal domain P satisfies* $\mathrm{rad}(P) = \hat{r} - 1$ *if and only if for all $i, i' \in \mathcal{H}(P) \cup \mathcal{V}(P)$ with $i \sqcap i'$ there exist $j, j' \in \mathcal{H}(P) \cup \mathcal{V}(P)$ with $j \sqcap j'$ such that $\Delta(i, j) \ge \hat{r}$ and $\Delta(i', j') \ge \hat{r}$. Otherwise,* $\mathrm{rad}(P) = \hat{r} - 2$.

With the above characterization, we can naively compute the diameter and the radius by checking all $O(n^4)$ quadruples $(i, i', j, j') \in \mathcal{H}(P) \times \mathcal{V}(P) \times \mathcal{H}(P) \times \mathcal{V}(P)$. However, the approach can be improved by using $\mathcal{G}(P)$.

▶ **Theorem 3.4.** *The rectilinear link diameter* $\mathrm{diam}(P)$ *and radius* $\mathrm{rad}(P)$ *of a rectilinear polygonal domain* P *consisting of* n *vertices and* h *holes can be computed in* $O(n^2 + nh\log h + \chi^2)$ *time, where* χ *is the number of edges of* $\mathcal{G}(P)$ *(i.e., the number of pairs of intersecting rectangles of* $\mathcal{H}(P)$ *and* $\mathcal{V}(P)$*).*

4 Computation via Matrix Multiplication

In this section we provide an alternative method to compute the diameter. Although not described here, a similar approach can also be used to compute the radius. This method also uses the condition in Theorem 3.2, but instead exploits the behavior of matrix multiplication on (0,1)-matrices. Recall that, given two (0,1)-matrices A and B, their product is $(AB)_{i,j} = \sum_k (A_{i,k} \cdot B_{k,j}) = |\{\, k\colon\ A_{i,k} = 1 \wedge B_{k,j} = 1 \,\}|$.

We define the (0,1)-matrices I, D and M. Note that we slightly abuse our notation and use i, j to indicate both matrix indices and their corresponding rectangles.

$$I_{i,j} = \begin{cases} 1 & \text{if } i \sqcap j, \\ 0 & \text{otherwise.} \end{cases} \qquad D_{i,j} = \begin{cases} 1 & \text{if } \Delta(i,j) = \hat{d}, \\ 0 & \text{otherwise.} \end{cases} \qquad M_{i,j} = \begin{cases} 1 & \text{if } (ID)_{i,j} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, matrix I indicates for each pair of rectangles if they properly intersect and have different orientations, whereas D indicates which rectangles are at oriented distance $\hat{d}$ from each other. For any entry in their product we then have

$$(ID)_{j,i'} = |\{\, j'\colon\ (j \sqcap j') \wedge (\Delta(i',j') = \hat{d}) \,\}|.$$

The matrix M then records which entries in (ID) are non-zero and we get

$$(DM)_{i,i'} = |\{\, j\colon\ D_{i,j} = 1 \wedge M_{j,i'} = 1 \,\}| = |\{\, j\colon\ \Delta(i,j) = \hat{d} \wedge (\exists_{j'}\colon\ j \sqcap j' \wedge \Delta(i',j') = \hat{d}\,)\}|.$$

So $(DM)_{i,i'} > 0$ if and only if there exists a pair j, j' for which $\Delta(i,j) = \hat{d}$, $j \sqcap j'$ and $\Delta(i',j') = \hat{d}$. By Theorem 3.2 if $(DM)_{i,i'} > 0$ and $I_{i,i'} = 1$ for any pair i, i', then the diameter is $\hat{d} - 1$ and otherwise it is $\hat{d} - 2$.

▶ **Theorem 4.1.** *The rectilinear link diameter* $\mathrm{diam}(P)$ *and radius* $\mathrm{rad}(P)$ *of a rectilinear polygonal domain* P *consisting of* n *vertices can be computed in* $O(n^\omega)$ *time.*

5 Computing the Diameter Faster

To test the condition of Theorem 3.2 we could simply iterate over each pair of rectangles i, j such that $\Delta(i,j) = \hat{d}$. For each such pair we could compute all pairs (i', j') such that $i \sqcap i'$ and $j \sqcap j'$ and test if $\Delta(i',j') = \hat{d}$. However, doing this naively may take $\Theta(n^4)$ time. Note however there are only $O(n^2)$ unique pairs (i', j') to test and regardless of which pair (i, j) was used to generate it, the diameter of P is $\hat{d} - 1$ if and only if at least one pair (i', j') has $\Delta(i',j') = \hat{d}$. We show how to more efficiently generate these pairs for the diameter. Unfortunately for the radius we must remember which pair (i, j) generates each pair (i', j') so this optimization doesn't work for the radius.

▶ **Theorem 5.1.** *The rectilinear link diameter* $\mathrm{diam}(P)$ *of a rectilinear polygonal domain* P *of* n *vertices can be computed in* $O(n^2 \log n)$ *time.*

Proof. *Sketch.* First, for each rectangle i, we find in $O(n \log n)$ time the set Q_i of rectangles at distance $\hat{d}$ from i. Then, using a ray-shooting data-structure by Giyora and Kaplan [6], we compute the set R_i which contains all rectangles j' that are orthogonal and intersect a rectangle $j \in Q_i$. We then store in a list for each rectangle in $j' \in Q_i$ the segment i. After doing this for each rectangle again iterate over all rectangles and use j' to denote the current rectangle. For each j', let $L_{j'}$ denote the set of rectangles i with $j' \in R_i$, which we stored in a list. Then using the same ray-shooting data structure we can compute in $O(n \log n)$ time the set $M_{j'}$ of all rectangles that are orthogonal to and intersect a rectangle $i \in L_{j'}$. Then we simply check every pair (i', j') with $i' \in L_{j'}$ and if any such pair is at distance $\hat{d}$ we report that the diameter is $\hat{d} - 1$. Since we iterate over all $O(n)$ rectangles twice and spend $O(n \log n)$ time on each of them the total running time is $O(n^2 \log n)$. ◀

References

1. Hee-Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh. A linear-time algorithm for the geodesic center of a simple polygon. *Discrete Comput. Geom.*, 56(4):836–859, 2016.
2. Sang Won Bae, Matias Korman, Yoshio Okamoto, and Haitao Wang. Computing the L_1 geodesic diameter and center of a simple polygon in linear time. *Comp. Geom.: Theor. Appl.*, 48(6):495–505, 2015.
3. Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. In *WADS*, pages 253–264, 2017.
4. Man-Kwun Chiu, Elena Khramtcova, Matias Korman, Aleksandar Markovic, Yoshio Okamoto, Aurélien Ooms, André van Renssen, and Marcel Roeloffzen. Rectilinear link diameter and radius in a rectilinear polygonal domain. *CoRR*, abs/1712.05538, 2017. URL: `https://arxiv.org/abs/1712.05538`, `arXiv:1712.05538`.
5. Hristo Djidjev, Andrzej Lingas, and Jörg-Rüdiger Sack. An $O(n \log n)$ algorithm for computing the link center of a simple polygon. *Discrete Comput. Geom.*, 8:131–152, 1992.
6. Yoav Giyora and Haim Kaplan. Optimal dynamic vertical ray shooting in rectilinear planar subdivisions. *ACM Trans. Algorithm.*, 5(3):28:1–28:51, 2009.
7. John Hershberger and Subhash Suri. Matrix searching with the shortest-path metric. *SIAM J. Comput.*, 26(6):1612–1634, 1997.
8. John Hershberger and Subhash Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.
9. François Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC*, pages 296–303, 2014.
10. Joseph S. B. Mitchell. An optimal algorithm for shortest rectilinear paths among obstacles. In *CCCG*, 1989.
11. Joseph S. B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8(1):55–88, 1992.
12. Joseph S. B. Mitchell, Valentin Polishchuk, and Mikko Sysikaski. Minimum-link paths revisited. *Comp. Geom.: Theor. Appl.*, 47(6):651–667, 2014.
13. Joseph S. B. Mitchell, Valentin Polishchuk, Mikko Sysikaski, and Haitao Wang. An optimal algorithm for minimum-link rectilinear paths in triangulated rectilinear domains. In *ICALP*, pages 947–959, 2015.
14. Bengt J. Nilsson and Sven Schuierer. Computing the rectilinear link diameter of a polygon. In *CG*, pages 203–215, 1991.
15. Bengt J. Nilsson and Sven Schuierer. An optimal algorithm for the rectilinear link center of a rectilinear polygon. *Comp. Geom.: Theor. Appl.*, 6:169–194, 1996.
16. Subhash Suri. On some link distance problems in a simple polygon. *IEEE T. Robotic. Autom.*, 6(1):108–113, 1990.